

**SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE  
STROJNÍCKA FAKULTA**

**MINIATURIZÁCIA EXPERIMENTU  
“MAGNETICKÁ LEVITÁCIA”**

**Bakalárska práca**

**SjF-13432-81742**

**2018**

**Jakub Mihalík**

**SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE  
STROJNÍCKA FAKULTA**

**MINIATURIZÁCIA EXPERIMENTU  
“MAGNETICKÁ LEVITÁCIA”**

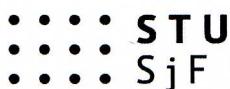
**Bakalárska práca**

**SjF-13432-81742**

Študijný odbor: 5.2.14 automatizácia  
Študijný program: Automatizácia a informatizácia strojov a procesov  
Školiace pracovisko: Ústav automatizácie, merania a aplikovanej informatiky  
Vedúci záverečnej práce: doc. Ing. Gergely Takács, PhD.  
Konzultant: Ing. Martin Gulan, PhD.

**Bratislava, 2018**

**Jakub Mihalík**



## ZADANIE BAKALÁRSKEJ PRÁCE

Študent: **Jakub Mihalík**  
ID študenta: 81742  
Študijný program: automatizácia a informatizácia strojov a procesov  
Študijný odbor: 5.2.14. automatizácia  
Vedúci práce: doc. Ing. Gergely Takács, PhD.  
Konzultant: Ing. Martin Gulan, PhD.  
Miesto vypracovania: ÚAMAI SjF STU v Bratislave

Názov práce: **Miniaturizácia experimentu "magnetická levitácia"**

Jazyk, v ktorom sa práca vypracuje: slovenský jazyk

Špecifikácia zadania:

Úlohou študenta je vytvoriť hardwarový a softwarový rámec na tvorbu modulu tzv. „magnetickej levitácie“ pre mikroradičovú prototypizačnú dosku Arduino (AVR). V rámci bakalárskej práce študent musí  
- navrhnuť experimentálne zariadenie, vybrať vhodné elektronické a mechanické komponenty, navrhnuť elektrické zapojenie a plošný spoj,  
- vyrobiť a dôkladne testovať kvalitu a funkčnosť 10 ks. zariadení,  
- napísať programátorské rozhranie (application programmers interface, API) pre ovládanie zariadenia,  
- vytvoriť inštruktažne príklady vhodné na didaktické nasadenie, napríklad identifikáciu sústavy alebo spätnoväzobné riadenie, pritom príklady musia názorne predvíeť funkčnosť zariadenia,  
- dôkladne zdokumentovať hardvér, softvér a príklady pomocou webstránky (Wiki) a podrobne komentovať zdrojový kód.

Rozsah práce: cca. 30-40 strán

Riešenie zadania práce od: 12. 02. 2018

Dátum odovzdania práce: 28. 05. 2018

**Jakub Mihalík**  
študent

prof. Ing. Cyril Belavý, CSc.  
vedúci pracoviska

**Slovenská technická univerzita**  
**v Bratislave**  
Dekanát Strojníckej fakulty  
Útvor pedagogických činností  
812 31 Bratislava, Nám. Slobody 17  
-3-

prof. Ing. Cyril Belavý, CSc.  
garant študijného programu

## **Čestné prehlásenie**

Vyhlasujem, že som záverečnú prácu vypracoval samostatne s použitím uvedenej literatúry.

Bratislava, 26. mája 2018

.....  
Vlastnoručný podpis

V prvom rade sa chcem podakovať vedúcemu práce doc. Ing. Gergelimu Takácsovi za umožnenie práce na projekte AutomationShield, ako aj za usmerňovanie a technickú výpomoc.

Ďalej by som sa rád podakoval kolegom, ktorí boli súčasťou tohto projektu, predovšetkým Tiborovi Konkolymu, ktorý mi v mnohých ohľadoch pomohol a ozrejmil spôsoby používania rôznych softvérów nevyhnutných pri návrhu, výrobe a popise mojej bakalárskej práce.

V neposlednom rade by som chcel podakovať rodine a priateľom za ich lajický, ale aj odborný pohľad, radu a rozhovory, ktoré mi pomohli viac porozumieť problematike obsiahnutej v tejto práci.

Bratislava, 26. mája 2018

Jakub Mihalík

**Názov práce:** Miniaturizácia experimentu “magnetická levitácia”

**Kľúčové slová:** magnetická levitácia, elektromagnet, shield Magneto, Arduino, spätnoväzbové riadenie, PID regulátor

**Abstrakt:** Práca sa zaoberá prevedením a využitím experimentu magnetickej levitácie pomocou elektromagnetu, na pedagogické účely. Postupne podľa kapitol je ozrejmené, čo pojem levitácia predstavuje, aké rôzne riešenia na realizáciu tohto experimentu existujú a podrobne je popísaný návrh a zostrojenie zariadenia Magneto - elektromagnetického levitátora v podobe shieldu pre prototypizačnú dosku Arduino UNO. Ďalej sa práca zaoberá vytvorením softvérového rozhrania na obsluhu Magneta vo forme knižnice. Uvedený je aj inštruktážny príklad využitia tohto rozhrania pri riadení procesu levitácie pomocou PID regulátora. Taktiež je uvedená charakteristika PID regulátora ako riadiaceho algoritmu. V závere práce sa nachádza konštruktívne zhrnutie funkčnosti celého experimentu. V prílohách sú uvedené kódy pre knižnicu, ako aj celý program inštruktážného príkladu.

**Title:** Miniaturization of the experiment “magnetic levitation”

**Key words:** magnetic levitation, electromagnet, shield Magneto, Arduino, closed-loop control, PID regulator

**Abstract:** This thesis presents the modeling of magnetic levitation experiment based on electromagnet and its possible application for educational purposes. The work describes the levitation effect, existing concepts of this experiment and our own realization of this experiment using Magneto device, which is an electromagnetic levitator in form of a shield for Arduino UNO. The paper also proposes library form software solution for Magneto usage. The thesis likewise presents an instructing example of utilization of the library for the control of the levitation by using PID regulator. The PID regulator characteristics as controlling algorithm are shown, as well. In the conclusion, the paper deals with outcomes of this experiment and its possible application. The library programs, as well as the instructing example program, are listed in the Appendix.

# Obsah

|  |           |
|--|-----------|
| <b>Zoznam obrázkov</b>   | <b>1</b>  |
| <b>Zoznam premenných a veličín</b>                               | <b>2</b>  |
| <b>Úvod</b>  | <b>3</b>  |
| <b>1 Elektromagnetický levitátor</b>                             | <b>4</b>  |
| 1.1 Mechanický princíp vznášania sa . . . . .                    | 4         |
| 1.2 Princíp levitácie za pomoci elektromagnetu . . . . .         | 4         |
| 1.3 Možnosti prevedenia elektromagnetického levitátora . . . . . | 5         |
| 1.3.1 Určovanie polohy optickou závorou . . . . .                | 6         |
| 1.3.2 Analógové určovanie polohy . . . . .                       | 6         |
| 1.4 Prevedenie levitátora Magneto . . . . .                      | 7         |
| <b>2 Schéma levitátora Magneto</b>                               | <b>8</b>  |
| 2.1 Elektromagnet . . . . .                                      | 9         |
| 2.2 Mosfet IRF520 . . . . .                                      | 11        |
| 2.3 PCF8591 . . . . .  | 13        |
| 2.4 Hallov senzor . . . . .                                      | 16        |
| 2.5 Arduino ako súčasť schémy . . . . .                          | 17        |
| <b>3 Plošný spoj PCB</b>   | <b>19</b> |
| 3.1 Časti PCB . . . . .  | 19        |
| 3.2 Dizajn PCB . . . . .   | 22        |
| 3.2.1 Nosné stĺpiky . . . . .                                    | 23        |
| 3.2.2 Hliníkový nosník . . . . .                                 | 23        |
| 3.2.3 Priestor pre levitáciu . . . . .                           | 24        |
| 3.3 Zhotovenie levitátora Magneto . . . . .                      | 24        |
| <b>4 Softvérové rozhranie levitátora Magneto</b>                 | <b>26</b> |
| 4.1 Knižnica . . . . .   | 26        |
| 4.1.1 Header . . . . .   | 27        |
| 4.1.2 Source . . . . .   | 28        |
| <b>5 PID</b>   | <b>33</b> |
| 5.1 Spätnoväzbové riadenie - PID regulátor . . . . .             | 33        |
| 5.2 Príklad použitia PID v levitátore Magneto . . . . .          | 35        |
| 5.3 Určovanie konštánt a práca s levitátorom Magneto . . . . .   | 39        |

|   |    |
|---|----|
| Záver                                       | 44 |
| Literatúra                                  | 46 |
| Dodatok A Zdrojový kód <i>header</i> súboru | 48 |
| Dodatok B Zdrojový kód <i>source</i> súboru | 49 |
| Dodatok C PID regulácia Magneta             | 51 |

# Zoznam obrázkov

|      |  |    |
|------|--|----|
| 1.1  | Silová rovnováha pri elektromagnetickej levitácií $\vec{F}_m = -\vec{F}_G; \vec{v} = 0.$ | 5  |
| 2.1  | Schéma elektromagnetického levitátora Magneto  | 8  |
| 2.2  | Elektromagnet.   | 9  |
| 2.3  | Elektromagnet typ P20/15.  | 11 |
| 2.4  | Princíp činnosti JFET-u  | 12 |
| 2.5  | Princíp činnosti MOSFET-u  | 13 |
| 2.6  | MOSFET IRF520  | 14 |
| 2.7  | DA prevodník PCF8591   | 14 |
| 2.8  | Princíp Hallovho javu.   | 17 |
| 2.9  | Hallov senzor A1302.   | 17 |
| 2.10 | Prototypizačná doska Arduino UNO.  | 18 |
| 3.1  | Návrh plošného spoja pre Magneto.  | 20 |
| 3.2  | 3D model navrhnutého PCB v programe DipTrace.  | 22 |
| 3.3  | Dôležité vzdialenosť pre upevnenie elektromagnetu.                                       | 22 |
| 3.4  | Kompletný levitátor Magneto spojený s Arduinom UNO.                                      | 25 |
| 5.1  | Schéma riadenia PID regulátorom  | 33 |
| 5.2  | USB osciloskop Analog Discovery 2.   | 40 |
| 5.3  | Postup určovania konštánt PID regulátora   | 42 |
| 5.4  | Detail vstupného a výstupného napäťa z Obr. 5.3(d).                                      | 43 |

# Zoznam premenných a veličín

| Symbol    | Premenná alebo veličina                 | Jednotka v sústave SI<br>(jednotka využívaná v práci) |
|-----------|---|---|
| $\vec{B}$ | vektor magnetickej indukcie             | T   |
| $D$       | vonkajší priemer elektromagnetu         | m (mm)  |
| $d$       | vnútorný priemer elektromagnetu         | m (mm)  |
| $e(t)$    | regulačná odchýlka v časovej oblasti    | závisí od veličiny                                    |
| $e(kT)$   | regulačná odchýlka v diskrétnej oblasti | závisí od veličiny                                    |
| $\vec{F}$ | vektor sily                             | N   |
| $H$       | výška elektromagnetu                    | m (mm)  |
| $I$       | elektrický prúd                         | A   |
| $i$       | premenná pre sumáciu                    | -   |
| $K_d$     | derivačná konštanta                     | -   |
| $K_i$     | integračná konštanta                    | -   |
| $K_p$     | proporcionálna konštanta                | -   |
| $k$       | premenná pre k-tu iteráciu              | -   |
| $L$       | dĺžka závitu                            | m (mm)  |
| $l$       | dĺžka                                   | m   |
| $n$       | počet závitov                           | -   |
| $\vec{p}$ | hybnosť                                 | kg m s <sup>-1</sup>                                  |
| $q$       | elektrický náboj                        | C   |
| $R$       | polomer                                 | m   |
| $\vec{r}$ | polohový vektor                         | m   |
| $r$       | veľkosť polohového vektora              | m   |
| $T$       | okamžitá hodnota času                   | s   |
| $T_d$     | derivačná časová konštanta              | -   |
| $T_i$     | integračná časová konštanta             | -   |
| $T_s$     | perióda vzorkovania                     | s ( $\mu$ s)  |
| $t$       | čas                                     | s   |
| $\tau$    | diferenciálne malí časový úsek          | s   |
| $U$       | napätie                                 | V   |
| $u(t)$    | vstup do sústavy v časovej oblasti      | závisí od veličiny                                    |
| $u(kT)$   | vstup do sústavy v diskrétnej oblasti   | závisí od veličiny                                    |
| $\vec{v}$ | rýchlosť                                | m s <sup>-1</sup>                                     |
| $w(t)$    | požadovaná hodnota regulovanej veličiny | závisí od veličiny                                    |
| $x$       | vzdialenosť                             | m   |
| $y(t)$    | výstup zo sústavy v časovej oblasti     | závisí od veličiny                                    |
| $y(kT)$   | výstup zo sústavy v diskrétnej oblasti  | závisí od veličiny                                    |

# Úvod

Dôležitou súčasťou v rámci výučby technických vied je možnosť experimentálne overiť pravdivosť naučenej problematiky. Študent je pri experimentoch konfrontovaný s reálnymi podmienkami a problémami, ktoré sú súčasťou každej praktickej úlohy, či už ide o školský projekt, zamestnanie alebo výskumnú činnosť. Tieto problémy sú iba ľahko opísateľné v rámci matematickej rovnice a zvyčajne sú nahradené konštantami špecifickými pre konkrétnu situáciu. Z tohto dôvodu je dôležité, aby študenti mali prístup k takýmto experimentom a zariadeniam sprostredkujúcim tieto edukačné možnosti.

Cieľom tejto bakalárskej práce je navrhnúť, zstrojiť a overiť funkčnosť takého zariadenia. Ide o magnetický levitátor s názvom Magneto, ktorý na vytvorenie magnetického poľa a tým na udržanie levitujúceho predmetu vo vzduchu používa elektromagnet. Takže hlavnou úlohou zariadenia je slúžiť na precvičovanie implementácie riadiacich algoritmov do reálneho dynamického systému.

Vývoj magneta je súčasťou školského projektu AutomationShiel, v rámci ktorého je vyvýjaných niekoľko takýchto zariadení, každé na inej fyzikálnej podstate.

Základnou snahou návrhu je tieto zariadenia zstrojiť v takej veľkosti, aby sa zmestili na plošný spoj, ktorý je možné nasunúť na prototypizačnú dosku (takéto plošné spoje sa nazývajú shielfy). Typ dosky, pre ktorý budú tieto shielfy vytvárané je Arduino UNO. Prototypizačnú dosku budú môcť študenti programovať a tým riadiť celé zariadenie. Študenti sa budú snažiť navrhnúť a napísať vhodný algoritmus pre riadenie týchto zariadení. V prípade Magneta sa budú snažiť dosiahnuť ustálené vznášanie sa predmetu vo vzduchu.

Pre zjednodušenie práce s týmito zariadeniami bude napísané aj softvérové rozhranie v podobe knižnice. V rámci tejto práce bude taktiež uvedený inštruktážny príklad, ako túto knižnicu používať pri riadení levitácie pomocou PID regulátora.

V tejto práci sa sice budem venovať iba PID regulátoru a jeho pomocou riadiť levitáciu, ale to nekladie nijaké prekážky využívaniu tohto zariadenia aj pri návrhoch komplexnejších systémov riadenia. Magneto sa bude dať použiť aj tréningu inicializácie sústavy alebo rôznych iných experimentoch spojených s riadením.

# 1 Elektromagnetický levitátor

## 1.1 Mechanický princíp vznášania sa

Vo všeobecnosti, levitátory sú zariadenia, ktoré sa snažia nejakým spôsobom kompenzovať všadeprítomnú tiažovú silu  $\vec{F}_G$ . Pomocou rôznych foriem energie vytvárajú a upravujú veľkosť "kompenzačnej" sily, smerujúcej opačným smerom ako  $\vec{F}_G$ .

Asi najlepším príkladom je helikoptéra, kde sa pomocou otáčania rotora, teda mechanickou energiou, vytvára vztlaková sila  $\vec{F}_{vz}$ . Čím rýchlejšie sa rotor otáča, teda je mu dodávaná väčšia energia, narastá aj sila opačná tiažovej. Nakoniec sa teleso dostáva do stavu, kedy je výslednica síl pôsobiacich na teleso nulová  $\vec{F}_{vz} = -\vec{F}_G$  (stav beztiaže).

Samozrejme táto podmienka pre levitáciu nestačí, respektíve náš lietajúci predmet by sa nikdy sám nevzniesol. Teleso si v tejto situácii iba zachováva svoj pohybový stav. Ostáva v pokoji alebo rovnomenom priamočiarom pohybe (Zákon zotrvačnosti). A keďže väčšina letov sa začína na zemi, kde je predmet v relatívnom pokoji, tak by sa len reakcia od zeme nahradila vztlakovou silou od rotora, no teleso by sa nepoholo.

Avšak, keď sú sily vyrovnané, aj malá sila v nejakom smere vie zmeniť tento pokojový stav na pohyb. Ak sa výkon rotora trochu zvýší  $|\vec{F}_{vz}| > |\vec{F}_G|$  helikoptéra okamžite začne stúpať.

Ak chceme stúpanie zastaviť, nestačí len ubrať výkon, tak aby sa sily rovnali. Z Newtonovho Zákona sily vieme, že sila je zmena hybnosti v čase.

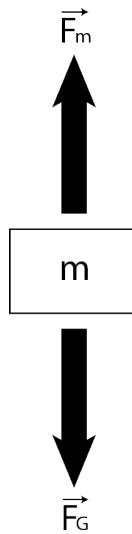
$$\vec{F} = \frac{d\vec{p}}{dt} = \frac{d}{dt}(m * \vec{v}) \quad (1.1)$$

Ak  $\vec{F} = 0$  znamená to, že hybnosť ostáva konštantnou a za predpokladu, že sa nemení hmotnosť telesa  $m$ , nezmení sa ani jeho rýchlosť  $\vec{v}$ . Z toho vyplýva, že je zapotreby sila, ktorá bude pôsobiť v protismere pohybu telesa, aby ho zastavila (v našom prípade musí platiť  $|\vec{F}_{vz}| < |\vec{F}_G|$ ) a v momente, keď bude rýchlosť predmetu nulová, treba vyrovnáť obe sily. Výsledný efekt je už zmienené vznášanie sa alebo levitácia.

I keď teoretický princíp vznášania sa je jednoduchý, práve vďaka platnosti Newtonovho zákona zotrvačnosti, nastoliť pokojový stav lietajúceho telesa v reálnych podmienkach je dosť zložité. Ako už bolo spomenuté, aj malá zmena jednej zo sôl tento stav narúša. Práve z tohto dôvodu je levitácia vysoko dynamický dej, u ktorého sa "kompenzačná" sila, ktorá zabezpečuje lietanie telesa, neustále mení.

## 1.2 Princíp levitácie za pomoci elektromagnetu

Hlavnou časťou tejto práce bolo vytvoriť levitátor, snažiaci sa dosiahnuť vyššie opísanú silovú rovnováhu za pomoci javov spojených s magnetizmom elektrického prúdu a mag-



Obr. 1.1: Silová rovnováha pri elektromagnetickej levitácii  $\vec{F}_m = -\vec{F}_G$ ;  $\vec{v} = 0$ .

netických vlastností látok. Sila, ktorá bude pôsobiť proti smeru tiažovej sily  $\vec{F}_G$ , bude miesto vztlakovej, sila magnetická  $\vec{F}_m$  (Obr. 1.1), ktorej velkosť bude riadená pomocou elektrickej energie vo forme elektrického prúdu.

Kedže sa jedná o magnetické javy, materiál levitujúceho predmetu musí byť schopný reagovať na tento druh síl. Takýmto typom materiálu je feromagnet (železo a niektoré zliaťiny), teda látka v ktorej, v dôsledku pôsobenia vonkajšieho magnetického poľa, dochádza ku silnej magnetizácii. Ďalšioím takýmto materiálom je permanentný magnet, generujúci vlastné magnetické pole, ktoré nie je podmienené prítomnosťou vonkajšieho poľa.

Kedže proces magnetickej levitácie v tejto práci bude dosahovaný za pomoci elektromagnetu a javov sním spojených, v ďalších častiach budem magnetickú levitáciu označovať výstižnejším názvom elektromagnetická levitácia (magnetickú levitáciu možno dosiahnuť aj inými spôsobmi ako elektromagnetom).

Ako som spomenul, ide o veľmi nestabilný proces, v ktorom je potreba rýchlych zmien síl pôsobiacich na lietajúce teleso. Táto skutočnosť je ešte umocnená tým, že veľkosť magnetickej sily (rovnako ako gravitačnej, či elektrickej) je závislá od štvorca vzdialenosťi oboch predmetov (pre podrobnejší opis elektromagnetickej sily a magnetických vlastností látok vid. [14, kapitola 8.4]).

Práve z tohto dôvodu je kocept elektromagnetickej levitátora, ako aj snaha o nastolenie silovej rovnováhy u vznášajúceho sa telesa, výborným príkladom na využitie riadiacich algoritmov v reálnom čase.

### 1.3 Možnosti prevedenia elektromagnetickej levitátora

Väčšina elektromagnetických levitátorov pozostáva z jedného elektromagnetu nachádzajúceho sa nad levitujúcim predmetom. V prípade takéhoto uloženia elektromagnetu a vznášajúceho sa telesa, magnetická sila  $\vec{F}_m$  predmet ťahá dohora. To zabezpečuje pomernú stabilitu a sily od okolia, pôsobiace v iných smeroch ako je smer  $\vec{F}_m$  alebo  $\vec{F}_G$ , môžme do značnej miery zanedbať.

Najdôležitejšou úlohou riadiaceho systému pri lietaní je vedieť v každom okamihu

presne určiť polohu vznášajúceho sa predmetu a na základe toho vyvodiť správny vstup do sústavy, ktorý ovplyvní sústavu požadovaným spôsobom. Pre elektromagnetickú levitáciu existujú dva základné spôsoby, ako túto polohu kontrolovať a z toho vyvodzovať vstupy.

### 1.3.1 Určovanie polohy optickou závorou

Najjednoduchším spôsobom je kontrola polohy predmetu pomocou "optickej závory" a ovládaním elektromagnetu štýlom ON-OFF. Zisťovanie polohy predmetu vykonáva dvojica zariadení - optický vysielač a prijímač (najčastejšie LED dióda a fototranzistor resp. fotorezistor). Umiestnením týchto dvoch zariadení proti sebe (tak aby žiarenie z vysielača dopadalo na prijímač) sa určí výška levitácie. Vznášajúci sa predmet je príťahovaný elektromagnetom vo vrchnej časti levitátora  $|\vec{F}_m| > |\vec{F}_G|$ . Akonáhle sa predmet dostane do výšky, kedy preruší optické spojenie, riadiaca jednotka (najčastejšie mikropočítač) zodpovedná za ovládanie elektromagnetu, elektromagnet vypne  $|\vec{F}_m| < |\vec{F}_G|$ . Teleso začne opäťovne klesať, až pokým neuvolní optickú závoru a elektromagnet sa zapne [6].

Tento spôsob riadenia je jednoduchý hlavne z dôvodu, že sa nesnaží o nastolenie a udržanie telesa v pokoji. Teleso je v skutočnosti neustále príťahované a spúštané k zemey elektromagnetom, v ktorom tečie buď maximálny alebo žiadny prúd. Dojem ustálenej levitácie vzniká vysokým vzorkovaním a rýchlosťou reakciou na stav optickej závory.

Pre rovnaký štýl riadenia, teda ON-OFF, možno použiť aj iné zariadenia tvoriace závoru. Jednou z možností je použitie switch senzoru Hallvho efektu. Jeho výstupom je digitálny signál, potvrdzujúci prítomnosť magnetického poľa o určitej indukcii - lietajúcim predmetom musí byť permanentný magnet<sup>1</sup>. Akonáhle sa dosiahne požadovaná indukcia magnetického poľa meraná hallovým senzorom (ekvivalent zatienenia optickej závory), elektromagnet sa vypne a lietajúci predmet začne padať. Keď indukcia klesne, elektromagnet sa znova zapne. Napriek požiadavke permanentného magnetu, je toto riešenie pomerne šikovné, lebo hallov senzor sa môže pripojiť priamo na elektromagnet (len v prípade, že hallov senzor je unipolárny - meria indukciu len jedného pólu, pre viac info viď Kap. 2.4). V prípade výmeny switch senzoru za analógový, toto riešenie ponúka ďalšie možnosti kontroly polohy predmetu.

### 1.3.2 Analógové určovanie polohy

Druhým spôsobom ako sa dá kontrolovať poloha levitujúceho predmetu je pomocou analógového výstupu, ktorým vieme určiť, nie len požadovanú polohu levitácie, ale aj kde sa predmet presne nachádza a ako je ďaleko od požadovanej polohy. Tento spôsob merania polohy poskytuje dostatočné množstvo informácií pre komplexnejšie programy riadenia než je už spomenutý ON-OFF systém. Silu elektromagnetu v tomto prípade môžme upravovať spojito pomocou veľkosti elektrického prúdu, ktorý ním preteká a tým sa snažiť dosiahnuť silovú rovnováhu  $\vec{F}_m = -\vec{F}_G$  v pokojovom stave telesa  $\vec{v} = 0$ .

Najčastejšie používaným zariadením pre zisťovanie polohy pri tomto typе elektromagnetickej levitácie je vďaka relatívne vysokej presnosti, nízkej cene a rýchlej reakcií na změnu polohy, hallov senzor s analógovým výstupom (používajú sa aj iné senzory a snímače, napr. senzory citlivé na svetlo ako v prípade zariadenia MAGLEV od firmy Quanser [12], ale sú podstatne drahsie).

---

<sup>1</sup>Teoreticky je aj možnosť použiť na lietanie feromagnet - napr. oceľové teleso, avšak jeho vlastné magnetické pole, vzniknuté pôsobením vonkajšieho poľa, je príliš slabé pre rádové hallové senzory. Pre praktické fungovanie by sa musel použiť veľmi silný elektromagnet

Na rozdiel od predchádzajúceho spôsobu riadenia, v tomto prípade však nie je nutnosťou, aby bol elektromagnet zavesený nad levitujúcim telesom. Niektoré prevedenia levitátorov majú elektromagnety umiestnené pod predmetom, ktorý je nimi tlačený. Táto verzia, si vyžaduje aby nadnášané teleso bolo v každom prípade permanentný magnet (u feromagnetu sa vždy vytvorí opačný pól ako je u elektromagnetu - pritahovali by sa). Je aj omnoho zložitejšia z dôvodu, že magnetická sila pôsobiaca na predmet, nemusí mať vždy smer opačný tiažovej sile a teleso sa môže veľmi ľahko vychýliť. Preto sú použité 4 elektromagnety, reguláciou ktorých je nadnášaný magnet vyvažovaný [8]. Taktiež nestačí len jeden hallový senzor, ale pre každý stupeň voľnosti riadenia treba jeden (uvažujeme 3 stupne voľnosti, v ktorých je možné sústavu ovládať - pohyb v osiach x, y, z). Z vyššie uvedeného vyplýva, že aj regulačný program bude podstatne zložitejší ako v prípade programu, ktorý rieši pohyb predmetu iba s jedným stupňom voľnosti riadenia (prípad zaveseného elektromagnetu a podním levitujúci predmet).

## 1.4 Prevedenie levitátora Magneto

Ako som sa už zmienil, systém riadenia ON-OFF a určovanie polohy závorou, sú sice funkčné spôsoby prevedenia, ale dosť jednoduché a neponúkajú žiadnu ďalšiu možnosť práce s ovládanou sústavou (teda lietajúcim predmetom).

Vzhľadom na to, že cieľom tejto práce je zostrojiť elektromagnetický levitátor, vo forme shieldu, bola snaha tento levitátor čo najviac miniaturizovať a zachovať možnosť ďalšej práce s ním. Z tohto dôvodu som sa rozhodol pre analógové určovanie polohy s tým, že elektromagnet sa bude nachádzať nad levitujúcim predmetom.

Kedže je snaha aby bol celý levitátor čo najmenší, rozhodol som sa, že na zisťovanie polohy predmetu použijem lineárnu hallovu sondu (viď Kap. 2.4). Tento senzor je dostačne malý a flexibilný, no pritom veľmi dobre reaguje na zmenu polohy predmetu. Na rozdiel od vyššie uvedených riešení, som hallovú sondu nepripevnil na elektromagnet, ale nachádza sa pod lietajúcim telesom v rovine plošného spoja. Toto uloženie je elegantnejšie a jednoduchšie na montáž.

Kedže som sa rozhodol použiť hallov senzor, levitujúcim predmetom musí byť permanentný magnet. Použitie permanentného magnetu má výhodu v tom, že jeho pole je silnejšie, ako pole iných feromagnetických látok. Vďaka tomu vie permanentný magnet (ďalej len magnet) vzlietnuť z väčšej vzdialenosťi od elektromagnetu. Vlastné pole magnetu taktiež poskytuje možnosť použiť menší elektromagnet (tým pádom aj slabší<sup>2</sup>).

Nevýhodou je ale neustála prítomnosť magnetického poľa. Väčšina elektromagnetov má oceľový obal, čo je feromagnetická látka. Ako náhle sa levitujúci magnet dostane príliš blízko elektromagnetu, medzi obalom a magnetom, od určitej vzdialenosťi, vznikne dostatočne silné pole, ktoré bude väčšie ako tiažová sila ťahajúca magnet k zemi. Magnet sa k obalu permanentne pritiaha, aj v prípade ak bude elektromagnet vypnutý. Tento problém je vyriešený tak, že magnet bude lietať v prieľadnej trubici, ktorá mu dovolí vyletieť len do takej výšky, aby pri vypnutom elektromagnete platilo  $|\vec{F}_m| < |\vec{F}_G|$ .

Napriek obmedzeniam spojených s prítomnosťou permanentného magnetického poľa som sa rozhodol pre neodimový magnet kvôli jeho veľmi silným magnetickým vlastnosťiam. Umožňuje to použitie magnetu menších rozmerov. Konkrétny magnet použitý v levitátore Magneto je neodimový disk priemeru 8 mm a hrúbky 2 mm.

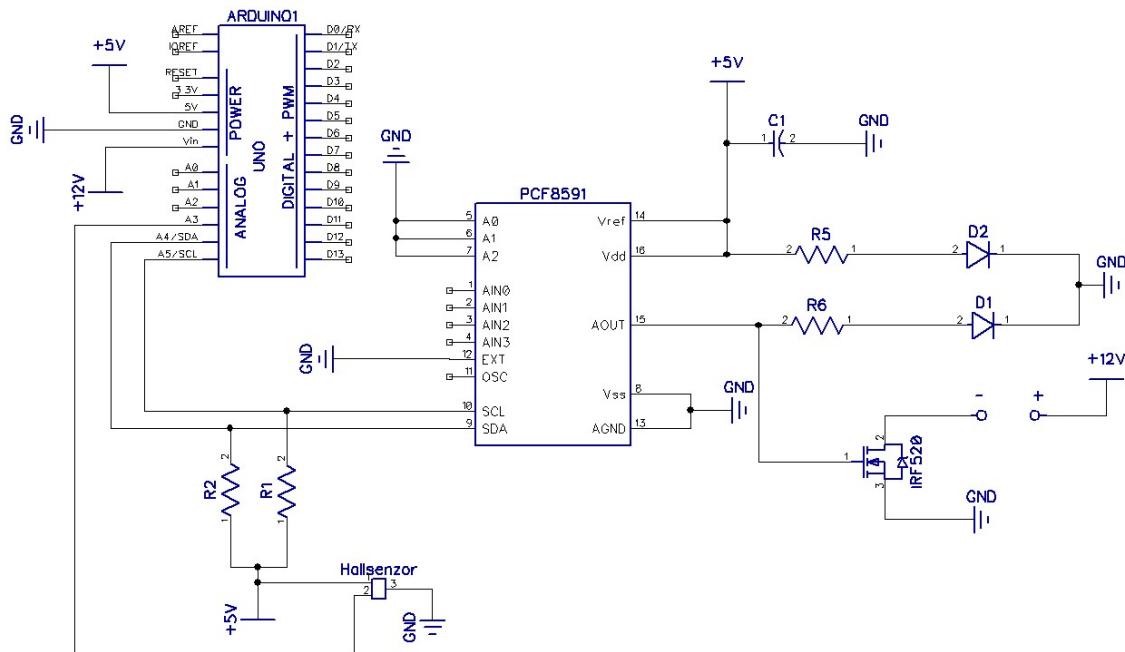
---

<sup>2</sup>Maximálna sila elektromagnetu závisí od najväčšieho možného pretekajúceho prúdu (priveľký prúd môže poškodiť cievku) a množstva závitov cievky v elektromagnete.

## 2 Schéma levitátora Magneto

V neslaedujúcej kapitole podrobnejšie opíšem jednotlivé funkčné časti mnou navrhnutého elektromagnetického levitátora Magneto (ďalej len levitátor Magneto alebo Magneto). Pokúsim sa objasniť, na akom princípe pracujú samotné súčasťky, akým spôsobom sú poprepájané a na základe toho obhájiť výber konkrétnych modelov.

Na Obr. 2.1 je zobrazený schématický nákres Magneta. Schéma je prvotným a základným návrhom celého zariadenia, objasňujúca vzájomné vzťahy medzi jednotlivými súčasťami. Na schématickom nákrese Magneta možno vidieť, že okrem elektromagnetu a hallovej sondy, ktoré som už vyššie spomenul, sa tu nachádza veľa ďalších nemenej dôležitých prvkov. Tieto zariadenia nie sú priamo zodpovedné za levitovanie, ale majú význam v napájaní a upravovaní napäťa pre spomenuté prvky alebo slúžia ako signalizačné časti, ktoré indikujú že zariadenie je v prevádzke.



Obr. 2.1: Schéma elektromagnetického levitátora Magneto

## 2.1 Elektromagnet

Srdcom celého levitátora je elektromagnet (Obr. 2.2). Skladá sa len z dvoch častí, z mednej cievky obmotanej okolo mageticky mäkkého materiálu. Teda je to naozaj jednoduché zariadenie, no má široké uplatnenie.



Obr. 2.2: Elektromagnet.

Kľúčovou vlastnosťou elektromagnetu je, že ku tvorbe magnetického poľa dochádza vplyvom toku elektrického prúdu. Z teórie elektrostatického poľa vieme, že medzi dvomi časticami s nábojom vzniká sila,

$$F_e = k \frac{q_1 q_2}{r^2} \quad (2.1)$$

, kde  $F_e$  je sila medzi nábojmi,  $q_1$  a  $q_2$  sú náboje častíc a  $r$  je vzájomná vzdialenosť oboch častic. Avšak skutočné pozorovania ukazujú že sila medzi časticami s nábojom je iná, ako vravý Coulombov zákon, čo do smeru aj do veľkosti. Táto sila sice vychádza z Coulombovho zákona, ale je rozšírená o zložku zohľadňujúcu rýchlosť oboch častic

$$\vec{F}_{12} = k \frac{q_1 q_2}{r_{12}^2} \vec{v}_1 \times (\vec{v}_2 \times \frac{\vec{r}_{12}}{r_{12}}) \quad (2.2)$$

, kde  $\vec{F}_{12}$  je magnetická sila akou pôsobí náboj  $q_1$  na  $q_2$ ,  $\vec{v}_1$  a  $\vec{v}_2$  sú vektory rýchlosťi oboch nábojov,  $\vec{r}_{12}$  je polohový vektor náboja 2 vzhľadom na 1 a  $r_{12}$  je veľkosť tohto vektora.

Konštantu  $k$  vystupujúca v oboch predošlých vzťahoch sa v sústave SI píše v tvare

$$k = \frac{\mu_0}{4\pi} \quad (2.3)$$

Z vyššie uvedeného vyplýva, že každá pohybujúca sa častica s nábojom vo svojom okolí vytvára vektorové pole, ktoré pôsobí na inú časticu s nábojom určitou silou (viď [14, str. 242]). Toto pole možno charakterizovať veličinou zvanou magnetická indukcia  $\vec{B}$ .

$$\vec{B} = \frac{\mu_0}{4\pi} \frac{q\vec{v} \times \vec{r}}{r^3} \quad (2.4)$$

$\mu_0$  je rozmerová konštanta, permeabilita vákuu - vyjadruje vplyv prostredia (v tomto prípade vákuu) na tvorbu magnetického poľa.

Princíp tvorby magnetického poľa tokom elektrického prúdu  $I$  vychádza práve z princípu usmerneného pohybu množstva takýchto častic s nábojom. Vzťahy pre magnetickú

indukciu sú pri každom tvari vodiča iné, lebo podľa Rov. (2.4) závisia od smeru pohybu častíc, takže aj od tvaru vodiča. Pre túto prácu je však zaujímavý jedine vzorec pre veľkosť magnetickej indukcie na osi solenoidu (dlhá cievka s množstvom kruhových závitov)<sup>1</sup>.

$$|\vec{B}| = \frac{\mu_0 n I}{2} \left[ \frac{x}{\sqrt{R^2 + x^2}} + \frac{l - x}{\sqrt{R^2 + (l - x)^2}} \right] \quad (2.5)$$

, kde  $n$  je počet závitov,  $l$  je dĺžka osi solenoidu,  $I$  je elektrický prúd,  $R$  predstavuje polomer solenoidu a  $x$  charakterizuje bod v ktorom meriame magnetickú indukciu.

Tento vzorec je súčasťou teoretického a nezohľadňuje vplyv reálneho prostredia na tvorbu magnetického poľa (ak je prostredím napr. feromagnet, veľkosť  $B$  sa zväčšuje), no podáva základnú informáciu o tom aké veličiny a akým spôsobom najviac ovplyvňujú magnetické pole solenoidu a teda aj elektromagnetu.

Ďalšia veľmi dôležitá informácia vypývajúca z Rov. (2.4) je, že veľkosť magnetickej indukcie  $\vec{B}$  je nepriamo úmerná štvorcu vzdialenosťi  $r$ . Takže ak meriame pole mimo solenoidu (no stále na jeho osi), čím budeme od neho ďalej, tým bude sila poľa kvadraticky klesať.

Tento fyzikálny popis som uviedol hlavne z toho dôvodu, že všetky spomenuté rovnice a z nich vyplývajúce vlastnosti solenoidu sa dajú s pár úpravami aplikovať aj na elektromagnet. Moja snaha bola zvoliť elektromagnet tak, aby bol dostatočne silný na zdvihnutie levitujúceho telesa, no zároveň dosť malý, aby splňal moju snahu miniaturizácie celého levitátora. Na základe vyššie uvedených zákonov a obmedzení som sa rozhodol pre elektromagnet model P20/15 (obr. 2.3) s vlastnosťami:

- Maximálne napájacie napätie  $U = 24$  V DC
- Maximálny prúd  $I = 0,25$  A
- Prídržná sila  $F = 25$  N
- Priemer  $D = 20$  mm
- Výška elektromagnetu  $H = 15$  mm

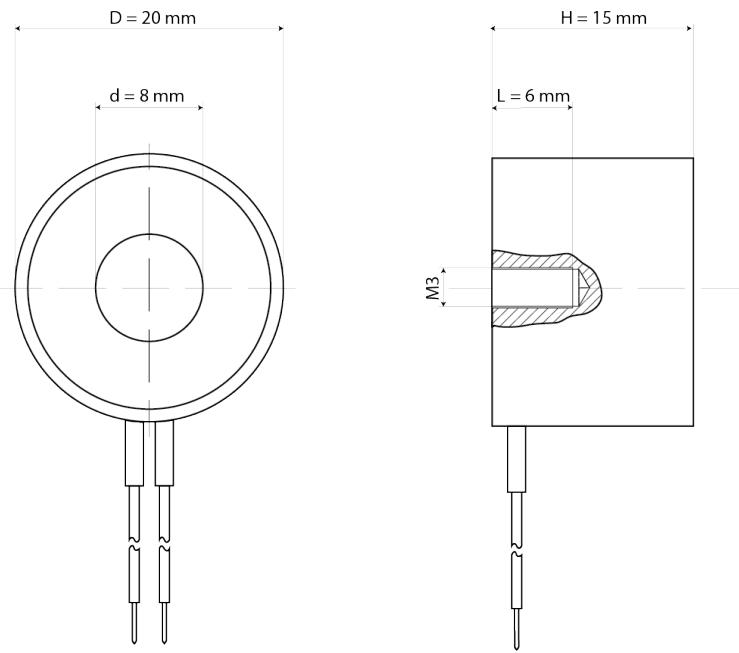
Napájacie napätie ktorým napájam elektromagnet je len 12 V. Je to z praktického dôvodu, že Arduino UNO môže byť napájané 12 V, ktoré možem využívať pomocou pinu Vin (Arduino UNO si pre ostatné periférie toto napätie transformuje na nižšie, lebo pracuje na 5 V TTL logike), tým pádom nepotrebujem ďalší zdroj napäcia špeciálne pre elektromagnet.

Ďalším dôvodom je, že pri vysokom napätií sa dosahujú vyššie prúdy a s tým spojené aj zahrievanie vodiča. Síce sa v dôsledku vyššieho prúdu dosiahne väčšia magnetická sila, no po určitom čase v dôsledku zahriatia poklesne. Dôvodom môže byť náras merného elektrického odporu [5] alebo pokles magnetických vlastností elektromagnetu [17]). Taktiež sa pri veľkom zahrievaní zvyšuje riziko popálenia.

Aj pri napájaní 12 V dochádza ku zahriatiu elektromagnetu na cca 50 °C, ale tento problém som vyriešil upevnením elektromagnetu na hliníkovú pásovinu, ktorá slúži ako

---

<sup>1</sup>Pre odvodenie vzťahu opisujúceho závislosť elektrického prúdu a magnetickej indukcie, ako aj výpočet magnetického poľa iných vodičov viď. [14, kapitola 6.1]



Obr. 2.3: Elektromagnet typ P20/15.

pasívne chladenie. Elektromagnet je k nej pripojený pomocou skrutky M3 (viac o upevnení elektromagnetu v Kap. 3.2).

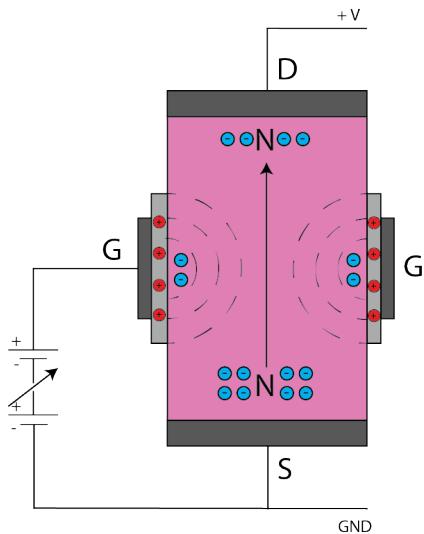
Ked' sa pozrieme na schému levitátora Magneto (Obr. 2.1), schématická značka elektromagnetu sa tam nenachádza. Miesto nej sú tam znázornené dva otvory "+" a "-", každý s inou polaritou. Pri elektromagnetoch je vemi dôležité dbať na to, aby boli do obvodu správne pripojené, lebo od toho závisí ich polarita, čo pri magnetickej levitácii, kde je levitujúcim predmetom permanentný magnet, môže silne ovplyvniť schopnosť magnetu lietať (v prípade zlého zapojenia sa magnet vôbec neodlepí od zeme, pre viac info viď Kap. 3.3).

## 2.2 Mosfet IRF520

V kapitole 1.3.2 som uviedol spôsob kontroly polohy, ktorý presne hovorí, v akej vzdialnosti od elektromagnetu sa nachádza levitujúci predmet. V prípade Magneta som na túto kontrolu použil lineárnu hallovú sondu, vďaka ktorej dostávam spojitý výstup zo sústavy. Avšak napájanie elektromagnetu je 12 V, elektromagnet má (odhliadnúc od už zmieneného zahrievania sa vodiča) konštantný odpor a tomu zodpovedá presný prúd. Otázkou je, ako viem vytvoriť na základe informácie z hallovho senzoru, spojitý vstup do sústavy, ktorým budem celý proces levitácie riadiť respektíve, ako dokážem vytvoriť rôzne hodnoty elektrického prúdu v intervale 0 až maximum pri napäti 0 až 12 V?

Jednou z možností riešenia je použitie tranzistoru typu FET (tiež unipolárny tranzistor), konkrétnie MOSFET.

Označenie tranzistoru FET pochádza z anglického "field effect transistor". Toto pomenovanie vystihuje základnú črtu, ktorou sa FET-y líšia od, v elektrotechnike možno známejších, bipolárnych tranzistorov. Ide o tranzistor riadený poľom. Na rozdiel od bipolárnych tranzistorov, kde polovodiče typu N (prevládajú voľné elektróny - nositeľ záporného náboja) a P (prevládajú diery - nositeľ kladného náboja) sú v priamom kontakte a



Obr. 2.4: Princíp činnosti JFET-u

priechodnosť PN prechodov je zabezpečená pomocou elektrického prúdu tečúceho medzi bázou a emitorom, u FET-ov elektrický prúd medzi gate a source (obdobu bázy a emitoru) vôbec netečie. Namiesto prúdu, priechodnosť FET-u ovláda napätie medzi týmito dvomi elektródami.

Poznáme niekoľko druhov FET-ov. Líšia sa materiálmi, ktoré sú v nich použité, ako aj princípom, akým napätie ovláda priechodnosť FET-u. Dva najznámejšie sú JFET a MOSFET. Oba typy môžu byť v prevedení s dvojakou polaritou:

- n - kanálový (obdoba NPN bipolárneho tranzistoru)
- p - kanálový (obdoba PNP bipolárneho tranzistoru)

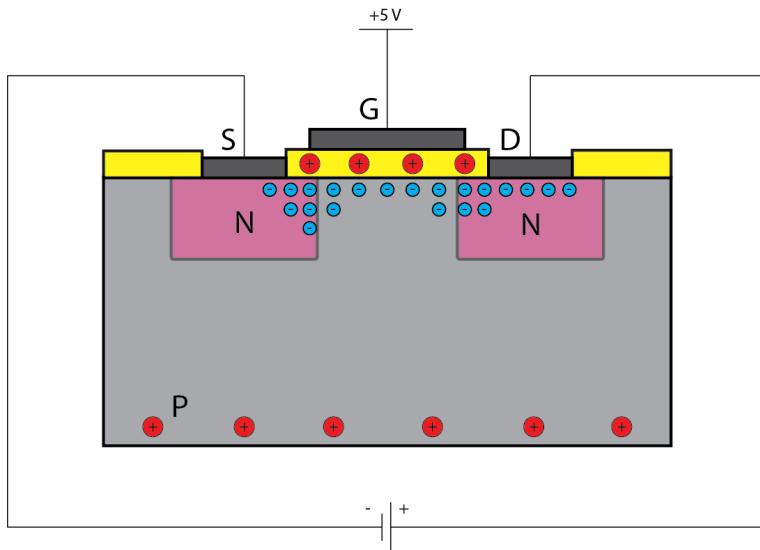
Pri JFET-e je vodivosť polovodičového hranola ovládaná riadiacou elektródou G - gate, na ktorú je privádzané napätie, ktoré s rastúcou veľkosťou "stahuje" voľné elektróny, aby neprechádzali zo zdrojovej elektródy S - source na koncovú elektródu D - drain. Rastúcim napäťím sa teda rozširuje plocha, do ktorej ked' sa dostane elektrón, nebude viest' prúd, ale pritiahe sa ku elektróde G<sup>2</sup> (viď obr. 2.4) [9].

Princíp MOSFETU je trošku iný. V polovodiči typu P (opäť uvádzam opis pre n-kanál) sú zasadené elektródy D a S z polovodiča typu N, no nie sú spolu prepojené. Medzi týmito dvomi elektródami sa nachádza riadiaca elektróda G, odizolovaná od polovodiča P tenkou vrstvou SiO<sub>2</sub> (z toho aj názov MOSFET - metal oxid semiconductor). Ako náhle na elektródu G priviedieme kladné napätie, voľné elektróny z D a S sú ľahšie odvodené ku kladnej elektróde G a tým vznikne vodivostný mostík voľných elektrónov, umožňujúci tok prúdu zo S do D. Bez prítomnosti napäťia na elektróde G by voľné elektróny neboli samé schopné prekonať bariéru v podobe PN prechodov (viď obr.2.5) [13].

Pri riešení návrhu levitátora Magneto som sa rozhodol, že pre ovládanie elektrického prúdu použijem MOSFET.

Základným dôvodom prečo som sa vôbec rozhodol pre FET miesto bipolárneho tranzistoru je ten, že Magneto bude riadený pomocou prototypizačnej dosky Arduino UNO,

<sup>2</sup>Tento opis zdrojovej a koncovej elektródy platí pre n-kanálový FET, pri p-kanáli je to naopak.



Obr. 2.5: Princíp činnosti MOSFET-u

kde viem ovládať napätie priamo (elektrický prúd len nepriamo). To mi poskytuje jednoduchšie ovládanie, ako aj lepší prehľad o tom, aká je práve “priechodnosť” obvodu, ako keby bol tranzistor ovládaný prúdom.

Dôvod prečo som uprednostnil MOSFET pred JFET je ten, že pri JFET-e ak na elektródu G neprivádzam napätie, obvod je otvorený, pri MOSFET-e v tejto situácii obvod zostáva uzavretý.

Konkrétny typ MOSFET-u ktorý používam je n-kanálový IRF520 (2.6). V schéme sa nachádza za pripojením pre elektromagnet. Jednotlivé elektródy sú tam označené číslami 1-gate G, 2-drain D, 3-source S (Pre kompletnejšiu charakteristiku IRF520 vid' [16]).

### 2.3 PCF8591

Z kapitolky 2.2 vieme, že na ovládanie veľkosti prúdu, ktorý tečie cez elektromagnet, musíme byť schopní vytvárať určitý rozsah napäti na riadiacej elektróde MOSFETU. Arduino UNO so svojím čipom ATmega380P nie je schopné vytvárať priamo analógové napätie. Je schopné tvoriť a prijímať digitálny signál na základe štandardu TTL s 5 V napájaním, prípadne prijímať analógový signál vďaka zabudovanému 10-bitovému AD prevodníku (analógovo-digitálny prevodník).

Spôsob akým sa dá vytvoriť aj iný napäťový výstup, ako je 0 alebo 5 voltov, pri takýchto mikročipoch sa nazýva PWM (Pulse-width modulation). V skratke, ide o vytváranie napodobeniny analógoého signálu pomocou skracovania a predlžovania vysokej napäťovej úrovne striedy počas jednej periody signálu (teda úprava času, v ktorom má signál 5 V v rámci jednej periody). Dá sa to predstaviť aj tak, že logickú 1 dostaneme, keď počas jednej periody budeme držať napätie na hodnote zodpovedajúcej zopnutému stavu a logicú 0 ak bude napätie počas jednej periody zodpovedať vypnutejmu stavu. PWM robí to, že napätie drží na logickej 1 iba časť periódy a následne padá na logickú 0. Pomerom ako dlho držíme vysoké či nízke napätie vieme v jednej periode vytvoriť priemerné hodnoty napäcia aj iné ako sú hodnoty zodpovedajúce 0 a 1.

Tento spôsob tvorby analógového signálu je spoľahlivý a vo väčšine prípadov, kedy potrebujeme generovať analógový výstup aj postačujúci. Avšak použitie PWM pri riadení



Obr. 2.6: MOSFET IRF520



Obr. 2.7: DA prevodník PCF8591

veľkosti elektrického prúdu na základe informácií získaných z hallového senzoru som experimentálne overil a nevyhovovalo požiadavkám kladeným na rýchlosť tvorby analógového signálu.<sup>3</sup>

Tento problém som vyriešil pomocou zariadenia DA prevodník (digitálno analógový prevodník), ktorý je schopný na základe digitálneho signálu pozostávajúceho z 1 a 0 vytvárať spojité napätie požadovanou rýchlosťou.

Konkrétny typ DA prevodníka, ktorý používam je 8-bitový čip PCF8591 od firmy NXP Semiconductors (Obr. 2.7). Tento čip má okrem funkcie DA prevodníka tiež možnosť AD prevodníka (v jednom prípade z digitálneho 8-bitového čísla vytvárame analógové hodnoty, v druhom prípade analógový signál zapisujeme do 8-bitovej správy). Túto funkciu však v Magnete nebudem potrebovať.

Tento mikročip (ako možno vidieť z obrázka 2.1) má 16 rôznych pinov.

Čip PCF8591 si vymieňa informácie pomocou komunikačného protokolu I<sup>2</sup>C. Tento protokol slúži na dorozumievanie sa viacerých zariadení pomocou zbernice s nadriadeným zariadením (v našom prípade s čipom ATmega380P na Arduine UNO). Zbernicu alebo v angličtine "bus" znamená prepojenie (v podobe vodiča), na ktoré môže byť pripojených naraz viacero rovnocenných zariadení. Zbernicou v protokole I<sup>2</sup>C sú práve piny SDA a SCL.

<sup>3</sup>Striedanie vysokého a nízkeho napäťia v rámci jednej periódy prebiehalo pre ovládanie elektromagnetu príliš pomaly. Namiesto analógového výstupu sa vytváral rýchli reťazec zysokého napätia (log 1) a nízkeho napätia (log 0).

Pri I<sup>2</sup>C sa jedná o synchrónnu komunikáciu, teda je zapotreby nejaký hodinový signál ktorý riadi pripojené zariadenia, kedy majú vysielať správy a kedy ich majú očakávať. Tento hodinový signál ide práve na pripojení SCL.

SDA slúži na oslovenie konkrétneho zariadenia pomocou adresy a priamo na výmenu informácií. Pomocou tohto spojenia vieme PCF8591 vysvetliť, či ideme používať jeho funkciu DA prevodníku alebo AD prevodníku.

Ďalšími pinmi sú A0, A1, A2, ktoré slúžia na zmenu adresy. Do čipu PCF8591 je totiž od výroby hardvérovo daná jeho adresa cez ktorú s ním nadriadené zariadenie nadviaže komunikáciu. Ak by sme však mali v obvode viacero takýchto čipov mohlo by dôjsť ku kolízii v komunikácii, kedže by všetky zariadenia boli oslovené naraz. Z tohto dôvodu sa dá pomocou týchto pinov, pripojením na zdroj napäťa alebo na zem, zmeniť adresu zariadenia (je to 7 bitové číslo a tieto piny nastavujú posledné 3 čísla na 0 alebo 1). Kedže v Magnete využijem len jeden PCF8591, všetky piny som uzemnil a tak nastavil na 0.

Piny AIN0 až AIN3 slúžia pri funkcií AD prevodníka ako analógové vstupy. Kedže ich nepoužívam, tak som ich nechal voľné. Ešte sa tu nachádza pin AGND, ktorý slúži na definovanie zeme ak by sa napätie privádzaného analógového signálu meralo voči inej zemi, ako je tá, ktorou je PCF8591T napájaný. Kedže sa nelisia a naviac ju nepoužívam, môžem ju pripojiť na zem napájania s označením V<sub>ss</sub>. Pre prevod analógového signálu na digitálny je dôležité tiež definovať referenčné napätie voči AGND, aby čip vedel ako má "rozkúskovať" analógové hodnoty na 8-bitové číslo. Túto možnosť čipu nevyužijem, ale pin V<sub>ref</sub> som pripojil na napájanie čipu.

Pin s najväčším významom pre Magneto má označenie AOUT. Jedná sa o vývod analógového signálu z DA prevodníka. Práve tento pin je pripojený na MOSFET IRF520, pomocou ktorého riadim elektrický prúd a vlastne aj celú elektromagnetickú levitáciu.

Napájanie čipu je riešené cez pin V<sub>dd</sub>, pričom napájacie napätie je od 2,5 V po 6 V. Okrem spomenutých pinov sa tu ešte nachádzajú dva a to EXT a OSC, ktoré ponúkajú ešte jednu možnosť použitia čipu a to ako zdroj hodinového signálu, čo tiež v levitátore Magneto pri mojej práci nepotrebujem (pre viac informácií o PCF8591 viď [11]).

V schéme Obr. 2.1 sa v blízkosti čipu PCF8591T nachádza niekoľko "sprievodných" súčiastok. Sú to štyri odpory, dve LED diódy a jeden kondenzátor. Tieto súčiastky sú pasívne súčasti Magneta. Ich význam spočíva v zjednodušovaní komunikácie a riadenia celého Magneta.

Prvé dva odpory o veľkosti 10 kΩ sú pripojené ku zberniciam SDA a SCL. Nazývajú sa tiež "pull-up" rezistory a ich význam spočíva v tom, že menia logické hodnoty stavov s napäťom a bez neho. Inak povedané, ak je napätie nulové, zodpovedá to logickej 0, a ak je napätie vysoké (v mojom prípade 5 V), to zodpovedá logickej 1.

Ďalšie dva odpory sa v obvode nachádzajú pripojené ku pinom V<sub>dd</sub> a AOUT. Majú veľkosť 1,2 kΩ a sú tam preto, lebo za nimi sa nachádzajú LED diódy. V prípade, že by tam odpory neboli, napätie 5 V, by ich mohlo spáliť. Tieto diódy slúžia na kontrolu obvodu, či ním naozaj tečie prúd. Prvá z diód má zelenú farbu, je pripojená k pinu V<sub>dd</sub> a svieti len v prípade, že je samotný čip napájaný požadovaným napäťom. Druhá dióda má červenú farbu a tým, že je pripojená k pinu AOUT, nám podáva vizuálnu informáciu o funkčnosti DA prevodníku (intenzita svetla LED diódy sa mení s veľkosťou napäťa

tvoreného DA prevodníkom).

Poslednou "sprievodnou" súčiastkou je kondenzátor s veľkosťou  $0,1 \mu\text{F}$ . Jeho úlohou v obvode je vyhľadzovať napájacie napätie.

## 2.4 Hallov senzor

O hallových senzoroch som niečo málo spomenul v Kap. 1.3. Ide o zariadenie schopné rozoznávať a prípadne aj určovať veľkosť magnetického poľa. Táto schopnosť je založená na elektromagnetickom jave zvanom Hallov jav. Ten vychádza z toho, akým spôsobom pôsobí magnetické pole na pohybujúci sa elektrón.

V Kap. 2.1 som uviedol vzťah, ako na seba vplývajú dva pohybujúce sa elektróny, Rov. (2.2) a Rov. (2.4) som definoval jednu zo zložiek tejto sily, ako magnetickú indukciu charakterizujúcu magnetické pole, vzniknuté pohybom jedného z nábojov.

Ak z Rov. (2.2) budeme považovať pohyb náboja  $q_1$ , za náboj vytvárajúci pole o indukcii  $\vec{B}$  v bode, v ktorom sa nachádza náboj  $q_2$  a ten sa hýbe rýchlosťou  $\vec{v}$ , môžme výslednú silu pôsobiacu na náboj náboj  $q_2$  zapísť ako

$$\vec{F}_m = q\vec{v} \times \vec{B} \quad (2.6)$$

, kde  $q$  predstavuje náboj  $q_2$  a  $\vec{v}$  je jeho rýchlosť.

Z tejto rovnice vidno, že vplyvom vektorových súčinov medzi rýchlosťou voľného elektrónu a magnetickou indukciou, je výsledná sila pôsobiaca na elektrón kolmá na rovinu definovanú týmito vektormi a upravuje pôvodnú rýchlosť a dráhu elektrónu (na Obr. 2.8 je táto "nová" rýchlosť znázornená ako  $\vec{v}_1$ ).

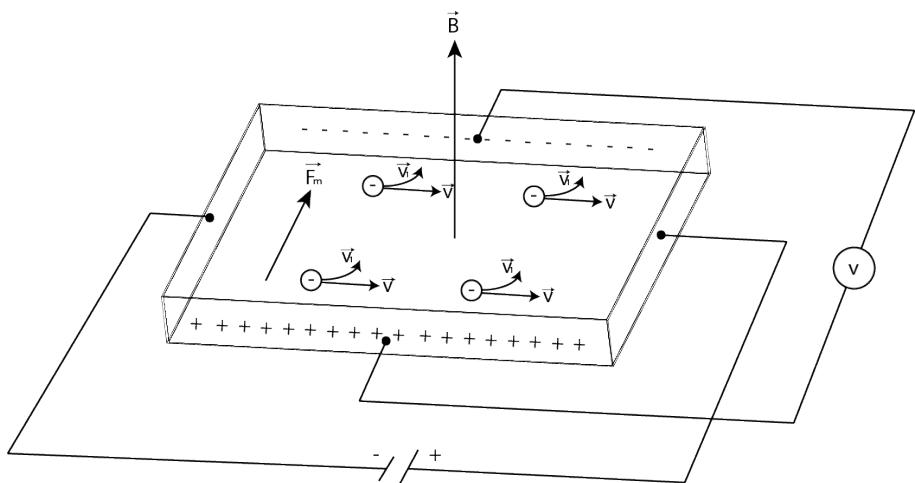
Predstavme si elektrický obvod, ktorého súčasťou je tenká vodivá platnička štvorcového tvaru. Táto platnička je pripojená do obvodu vodičmi pripájkovanými k jej dvom protiľahlým stranám. Na ďalšie dve protiľahlé strany je pripojený voltmeter. Ak v obvode s platničkou bude tieť elektrický prúd, zatiaľ voltmeter nič nenameria.

Teraz si predstavme že celý obvod vložíme do homogénneho magnetického poľa tak, že vektor magnetickej indukcie  $\vec{B}$  bude kolmý na vodivú platničku. Zrazu môžme pozorovať jav, pri ktorom na voltmetri nameriame napätie. Na základe Rov. (2.6), na pohybujúce sa elektróny pôsobí magnetická sila, ktorá ich tlačí na jednu zo strán doštičky. To spôsobuje nadmerný záporný náboj na jednej strane a kladný náboj znova na druhej, vďaka čomu vieme na týchto protiľahlých stranách nameriť napätie (Obr 2.8). Vďaka tomuto princípu sme schopní relatívne ľahko a presne merať magnetické pole [14, Kapitola 8.5.2].

Hallové sondy majú tri piny. Dva sú zodpovedné za napájanie, teda vytvárajú obvod v ktorom je pripojená platnička a tečie ním elektrický prúd. Na treťom pine sa tvorí analógový signál a viem na ňom nameriť napätie. Ako som spomíнал levitujúcim telesom bude permanentný magnet. Jeho polohu budem zistovať pomocou nameraného napäťia na hallovom senzore. Čím bude magnet bližšie ku senzoru, tým vciajce sa elektróny vychýlia a nameriam vyššie napätie.

Okrem samotnej polohy levitujúceho predmetu sa dá pomocou hallovej sondy zistiť aj polarita natočenej strany magnetu ku sonde. Takéto hallové sondy schopné rozlišovať póly magnetu a vytvárať na základe toho analógovú informáciu sa nazývajú bipolárne.

Veľkosť magnetickej sily pôsobiacej na pohyblivé elektróny je kvadraticky závislá od vzdialenosťi (vyplýva z Rov. (2.4)), teda aj vzniknuté napätie by sa so vzdialenosťou menilo kvadraticky, čo by mohlo komplikovať výpočty na základe nameraného napäťia. Preto



Obr. 2.8: Princíp Hallovho javu.



Obr. 2.9: Hallov senzor A1302.

existujú takzvané lineárne hallové sondy, ktoré vo svojom obvode obsahujú už zakomponované linearizačné prvky.

Pri návrhu levitátora Magneto som použil hallovú sondu od firmy Allegro, A1302, ktorá je lineárna a bipolárna (Obr. 2.9). To, že je sonda lineárna je výhodou pre levitátor z dôvodu jednoduchšej manipulácie s nameranými hodnotami a na základe toho zisťovaním polohy magnetu. Bipolarita je výhodou v tom, že jednak to zjednodušuje prípájanie sondy na levitátor (je jedno ktorou stranou je natočená) a teoreticky poskytuje možnosť prepílovať elektromagnet a kontrolovať polohu levitujúceho magnetu bez ohľadu na orientáciu pôlov.

Tento senzor sa dodáva v dvoch baleniach a to v podobe tenkej doštičky (3x4 mm) s pinmi vzchádzajúcimi z jednej strany, alebo v podobe SMD súčiastky s označením A1302ELHLT-T. Síce na prvotnom prototype Magneta sa nachádza hallov senzor v prvom spomenutom prevedení, vo finálnej verzii sa uvažuje SMD prevedenie z dôvodu menšieho rozmeru (vyhovuje snahe miniaturizovať levitátor) a aj elegantnejšieho pripojenia ku plošnému spoju Magneta. Na Obr. 2.1 sú piny hallovho senzoru označené číslami 1-napájanie 5 V, 2-analógový výstup, 3-napájanie zem (pre kompletnejšiu charakteristiku A1302 viď [1]).

## 2.5 Arduino ako súčasť schémy

Magneto je zariadenie, ktoré nie je schopné fungovať samostatne. Pre svoju prevádzku potrebuje nejaké iné “inteligentnejšie” zariadenie, ktoré bude riadiť celú levitáciu a bude mu dodávať energiu. V mojom prípade týmto zariadením je prototypizačná doska Arduino

UNO (Obr. 2.10).

Arduino UNO obsahuje množstvo komponentov a funguje ako samostatné zariadenie. Jeho hlavnou úlohou je pomocou všetkých svojich periférií zjednodušiť prístup k práci s mikročipom ATmega328P. Väčšina spojení Arduina a iných zariadení, ktoré má ovládať, je realizovaných cez rôzne piny. Tieto piny sprostredkúvajú jednotlivé komunikačné periférie, s ktorými dokáže mikročip pracovať. Arduino UNO (ako aj všetky ostatné prototypizačné dosky) má polohu a typ pinov pevne dané a práve z tohto dôvodu sa v schéme na Obr. 2.1 nachádza komponent Arduino.

V tomto prípade sa nejedná o celú dosku Arduino UNO, ale len o komponent naznačujúci, aké jednotlivé periférie bude mať levitátor Magneto k dispozícii. Taktiež, vďaka prítomnosti tohto komponentu v schéme, pri návrhu plošného spoja odpadáva problém s umiestnením pinov pre Magneto (treba ich geometricky presne osadiť, aby zapadali do prototypizačnej dosky - toto geometrické osadenie rieši práve komponent Arduino).

Pri prepojení Magneta a Arduina UNO bude k dispozícii 14 digitálnych pinov (môžu slúžiť aj na tvorbu PWM signálu), 6 analógových z čoho 2 sa dajú využiť na vytvorenie I<sup>2</sup>C komunikácie a 8 pinov zodpovedajúcich za napájanie (v schéme ich je menej lebo piny GND sú vzájomne prepojené).

Levitátor Magneto využíva na komunikáciu len 3 analógové piny. Jeden na prenos inforácie z hallovho senzoru a dva na I<sup>2</sup>C komunikáciu, ináč nazývané aj SDA a SCL.

Kedže levitátor Magneto nemá vlastný zdroj energie, okrem komunikácie som cez Arduino UNO riešil aj napájanie levitátora. Ako je vidieť zo schémy Obr. 2.1 na funkčnosť zariadenia potrebujem 5 V na napájanie hallovho senzoru a DA prevodníku PCF8591T a 12 V na napájanie elektromagnetu. Arduino UNO síce pracuje na 5 V TTL logike, ale v prípade, že je napájané 12 V zdrojom napäťia cez barrel jack, sme schopní cez pin Vin z neho odoberať 12 V, bez toho aby to nejakým spôsobom poškodilo alebo obmedzilo funkčnosť zvyšku prototypizačnej dosky.



Obr. 2.10: Prototypizačná doska Arduino UNO.

# 3 Plošný spoj PCB

Schématický návrh, ktorým som sa zaoberal v predchádzajúcej kapitole, určuje, ako bude levitátor fungovať a aké budú vzťahy medzi jednotlivými komponentami, ktoré ho tvoria. Ďalšou úlohou pri návrhu levitátora je na základe schémy vytvoriť model plošného spoja (ďalej PCB z anglického “printed circuit board”), ktorý bude definovať, kde sa budú spomínané komponenty nachádzať a bude určovať dizajn levitátora.

Ako som spomínal, Magneto bude navrhnutý v podobe shieldu určeného pre prototypizačné dosky Arduino UNO. Tento fakt je zásadný pri grafickom návrhu shieldu, lebo typ dosky pre ktorú je určený, je rozhodujúci pri definovaní polohy jednotlivých komponentov (najme pinov).

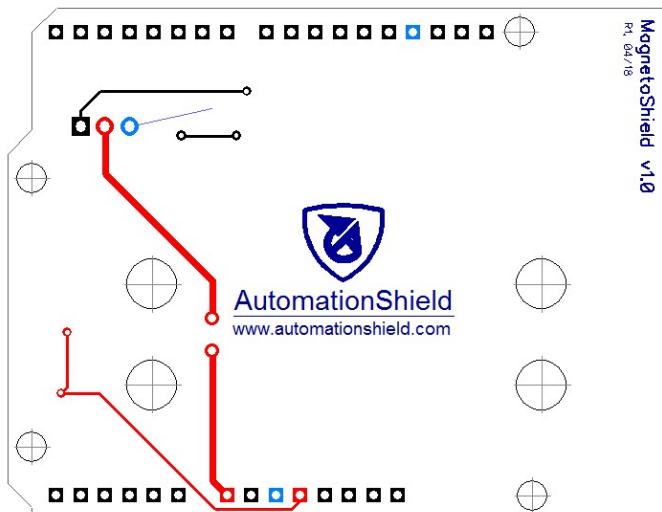
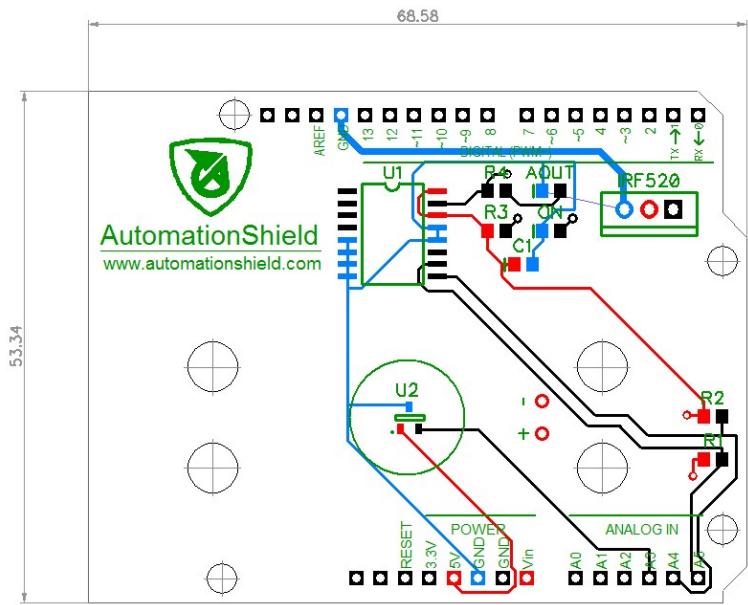
Schému levitátora Magneto som robil v programe DipTrace. Tento program ponúka možnosť každej schématickej súčiastke priradiť, ako bude vyzerat pri návrhu PCB. Po vytvorení schémy sa zadá “convert to PCB” a program vygeneruje všetky komponenty v 2D podobe, v akej budú naznačené na PCB. Tým, že v schéme na Obr. 2.1 sa nachádza komponent Arduino (viď Kap. 2.5), jednou z vygenerovaných častí sú geometricky definované okraje dosky s pinmi. Všetky ostatné komponenty musia zpadať do priestoru ohraničeného týmito okrajmi (viď Obr. 3.1).

## 3.1 Časti PCB

Pri návrhu PCB má každý z použitých komponentov priradenú svoju značku, ktorá ho na Obr. 3.1 reprezentuje zelenými okrajmi. Tieto ohraničenia iba pomáhajú Lahšie si predstaviť, ako bude daný komponent na PCB vyzerat, prípadne sa môžu použiť ako vodiace čiary pre komponenty, ktoré sa pridajú na PCB neskôr a nebudú prepojené s ostatnými súčasťami pomocou vodivých spojov. Takýmto komponentom je izolačný krúžok a priesvitná trubica, ktoré ozrejmí neskôr (Kap. 3.2.3).

Taktiež majú význam pri určovaní, ktorý komponent kam patrí. Z tohto dôvodu má každá značka svoj názov (s výnimkou kruhu pre lepenie izolačného krúžku). V prípade, že na PCB sa nachádza viacero komponentov v rovnakom prevedení (na pohľad vyzerajú rovnako ale v parametroch a účele sa môžu lísiť), názvy značiek nám zjednodušujú umiestnenie súčiastok pri spájkovaní, prípadne nám hovoria, na akom mieste je ktorý komponent (funkcia súčiastky nemusí byť na prvý pohľad zrejmá).

Ako som spomenul, zelené okraje súčiastok sú len pre orientáciu. Pri generovaní PCB návrhu sa každej súčiastke okrem značky priradia aj medené plôšky, ktoré budú súčiastku spájať so zvyškom obvodu. Voľba rozmerov týchto plôšok, ako aj rozostup medzi nimi alebo ich typ (može byť to medená plocha alebo diera s plochou okolo), závisia od prevedenia konkrétnej súčiastky. Množstvo súčiastok sa totiž vyrába v rôznych prevedeniach. Funkcia, ako aj parametre súčiastky sú rovnaké, ale to kde majú vyvedené piny alebo aké majú



Obr. 3.1: Návrh plošného spoja pre Magneto.

rozmery sa môže lísiť.

Pri návrhu PCB to bola jedna z najdôležitejších vecí, ktoré som musel zohľadniť. Po výbere vhodných komponentov bolo treba určiť aj v akom verzií budú najvhodnejšie pre levitátor Magneto. Na základe snahy levitátor čo najviac miniaturizovať som sa rozhodol pre komponenty v prevedeniach SMD (z angličtiny “surface mount devices”), jednak kvôli ich malým rozmerom a jednak sa na PCB pripájajú pomocou medených plôch, nevyžaduje sa pre súčiastku diera.

Jedinou súčiastkou, ktorá nie je SDM, je MOSFET IRF520 z dôvodu, že táto súčiastka sa nevyrába v tomto zmenšenom prevedení. Ako možno vidieť z Obr. 3.1, IRF520 sa ku doske prispájkuje pomocou predvŕtaných dier s medeným okrajom. Rovnakým spôsobom sa k doske pripájajú aj piny, ktoré budú shield Magneto spájať s prototypizačnou doskou Arduino UNO.

Jednou z výhod PCB oproti prototypom, v ktorých sú komponenty pospájané káblami, je stabilné vodivé prepojenie medzi súčiastkami. Toto prepojenie je realizované pomocou vodivých medených cestičiek, ktoré sú od výroby súčasťou PCB.

Na Obr. 3.1 majú tieto cesty čiernu červenú a modrú farbu. Toto zafarbenie nemá žiadny vplyv na výrobu konkrétneho spoja (na vyrobenej doske nebude vidno farebný rozdiel). Použil som ho z dôvodu, aby bolo ľahšie rozlíšiteľné, ktoré cesty budú napájať súčiastky (červená), ktoré budú vytvárať zem (modrá) a v ktorých cestách bude prebiehať komunikácia. Toto rozlíšenie zjednodušuje následnú kontrolu, či je obvod postavený logicky a súčiastky sú pripojené na správne piny shieldu.

Čo sa týka hrúbky ciest, na obrázku si možno všimnúť, že sa tam nachádzajú dve cesty s väčšou hrúbkou ako majú ostatné. Taktiež sú pripojené na samostatný zdroj napäťia (Vin) a samostatnú zem (GND). Je tomu tak preto, lebo tento spoj vytvára napájanie elektromagnetu (zapojený do obvodu v mieste “+” a “-”) a tadiaľto bude tieť vyšší prúd<sup>1</sup>.

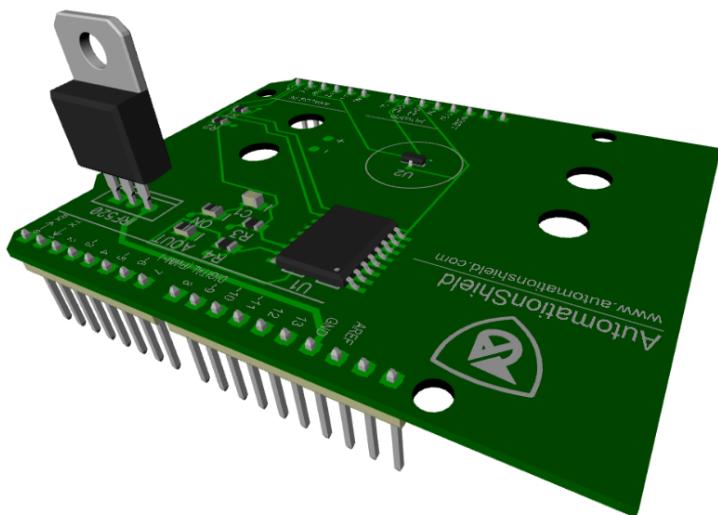
Ako je vidieť, návrh PCB pre levitátor Magneto využíva na prepájanie komponentov obe strany dosky. Je to hlavne z dôvodu, že cesty sa nesmú krížiť a čím viacej komponentov doska obsahuje, tým je vytvorenie takejto schémy ciest problematickejšie. Keď využívame obe strany dosky, výrazne to pomáha pri návrhu ciest a vo výsledku to aj znižuje cenu, lebo cesty sa môžu ľahšie priamejšie a tým pádom nie je za potreby toľko medi (samozrejme dopad na cenu je zrejmý až pri objednávkach veľkých počtov kusov).

Pripojenie MOSFETU IRF520 je realizované pomocou dier, tým pádom využitie vodivých ciest na spodnej strane PCB je možné od pripojenia súčiastky k shieldu. U SMD súčiastok to však hned možné nie je, kedže sú pripájané len povrchovo a to na vrchnú stranu PCB. Na preklopenie cest z jednej strany na druhú sa používajú maličké diery, ktoré je možné vidieť napr. v spodnej časti Obr. 3.1, pri súčiastkách s označením R1 a R2 (rezistory o veľkosti 10 kΩ).

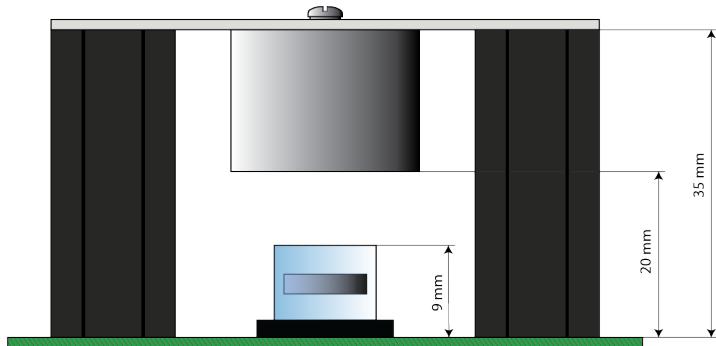
Posledná časť návrhu z Obr. 3.1, ktorú som ešte nespomenul, sú diery na upevnenie. Na Magnete sú dve rôzne veľkosti dier. Prvé 4 diery, ktoré lemujú okraj dosky, tam sú z dôvodu, ak by bolo potrebné dosku k niečomu pripojiť. Tieto diery sú robené pre skrutky M3 a sú zvyklosťou používanou na prototypizačných doskách a shieldoch. Pri návrhu Magneta som neuvažoval s využitím týchto dier, ale ponúkajú širšie možnosti v prípade ďalšej práce so shieldom.

Druhý typ dier ktorý sa na doske nachádza, sú väčšie diery, tiež určené pre skrutkový

<sup>1</sup>Ak by bol vodič príliš tenký, v dôsledku väčšieho prúdu by sa mohol ľahko prehriať a spoj by sa poškodil.



Obr. 3.2: 3D model navrhnutého PCB v programe DipTrace.



Obr. 3.3: Dôležité vzdialenosť pre upevnenie elektromagnetu.

spoj. Veľkosť a umiestnenie týchto dier, som na rozdiel od predošlého typu, určoval sám. Tieto diery budú slúžiť na uchytenie dištančných stĺpikov, ktoré budú niesť elektromagnet (viac v Kap. 3.2.1). Veľkosť dier (robené pre skrutku M5), ako aj ich vzájomné postavenie sú urobené na základe parametrov týchto stĺpikov.

## 3.2 Dizajn PCB

Po kompletizácii všetkých prvkov, ktoré má shielad Magneto obsahovať, som vytvoril 3D model jeho plošného spoja Obr. 3.2.

V predošlých častiach som spomíнал jednotlivé elektronické zariadenia, opisoval som ich vlastnosti a spôsoby, akým sú poprepájané a podieľajú sa na tvorbe Magneta. Teraz by som rád priblížil tie časti Magneta, ktoré samé o sebe s elektromagnetickou levitáciou nesúvisia, ale tvoria kostru tohto zariadenia a prípadne pomáhajú pasívnym spôsobom vylepšovať samotnú levitáciu.

### 3.2.1 Nosné stĺpiky

Pri experimentálnych pokusoch s elektromagnetom a permanentným magnetom som zistil, že naplno zopnutý elektromagnet je schopný pritiahnúť permanentný neodymový magnet zo vzdialenosťi približne 20 mm. Výška použitého elektromagnetu je 15 mm. Z týchto rozmerov som zistil, že elektromagnet musím upevniť vo výške 35 mm na PCB (viď Obr. 3.3).

Možnosti ako upevniť elektromagnet v tejto výške je veľmi veľa. Pôvodne som uvažoval použiť na zdvihnutie elektromagnetu hliníkové profily, ktoré by bolo potrebné presne narezať a k PCB by sa pripevnili pomocou lepidla.

Zmysel návrhu elektromagnetického levitátora Magneto však spočíva v tom, aby celé zariadenie bolo čo najmenšie, no zároveň postavené z čo najväčšieho množstva sériovo vyrábaných súčiastok.

Nakoniec som sa pri návrhu rozhodol pre dištančné šestihanné stĺpiky vyrobené z polyamidu<sup>2</sup>. Tieto stĺpiky majú výšku 35 mm od výroby. Ďalšou ich prednosťou oproti pôvodne uvažovanému návrhu je, že k doske sa pripájajú skrutkou. Vo vrchnej aj spodnej časti sa nachádza diera s metrickým závitom M5 o dĺžke závitu 10 mm (M5x10). To poskytuje možnosť stĺpiky v prípade potreby demontovať alebo vymeniť niektoré časti levitátora.

Stĺpiky sa vyrábajú v tejto výške s rôznymi otvormi pre skrutku (od M2 po M6), čo ovplyvňuje šírku stĺpika. Kedže skrutke M5 zodpovedá stĺpik o šírke najkratšej strany 10 mm (u šestuholníka je to vzdialenosť dvoch protiľahlých strán), tak som sa rozhodol použiť dva takéto stĺpiky vedľa seba. Použitie dvoch stĺpikov jednak poskytuje väčšiu stabilitu a jednak pôsobí estetickejšie, kedže hliníkový profil s elektromagnetom, ktoré budú priskrutkované ku stĺpikom zhora, majú šírku 20 mm. Ak by bol len jeden stĺpik, profil by po stranách prečnieval.

### 3.2.2 Hliníkový nosník

Po vymyslení spôsobu ako dostať elektromagnet do požadovanej výšky som riešil otázku, akým spôsobom elektromagnet ku stĺpikom upevním.

Elektromagnet s priemerom 20 mm má vo svojej vrchnej časti predvŕtanú dieru M3x6 (viď Obr. 2.3), tým pádom najvhodnejším profilom na uchytenie by bola tenká priečka z nemagnetického materiálu o šírke 20 mm.

Pozorovaniami som však prišiel k záveru, že elektromagnet sa prechodom elektrického prúdu citelne zahrieva a to aj napriek tomu, že použitý elektromagnet je stavaný na napätie 24 V (použité napätie u levitátora Magneto je 12 V). Toto zahrievanie, ako som spomíнал v Kap. 2.1, môže nepriaznivo pôsobiť na magnetické vlastnosti elektromagnetu. Pri experimentoch bola dosiahnutá teplota až okolo 50 °C. Na základe tohto som sa rozhodol, že elektromagnet upevniť ku stĺpikom pomocou tepelne vodivého materiálu, ktorý by pôsobil ako chladič. Najvhodnejší materiál splňajúci podmienku nízkych magnetických vlastností a vysokej tepelnej vodivosti je hliník, konkrétnie hliníková pásovina hrúbky 2 mm. Tento profil sa ale zvyčajne predáva len vo väčších rozmeroch a jeho úpravu na nami požadovanú dĺžku 50 mm, ako aj vyvŕtanie dier pre upevnenie elektromagnetu a priskrutkovanie ku stĺpikom, je potrebné vykonať dodatočne. Predpripravené profily sa veľkovýrobne nepredávajú a ich zákazková výroba je v malom počte neprimerane drahá.

<sup>2</sup>Materiál použitý pri upevnení elektromagnetu k doske musí mať čo najmenšie magnetické vlastnosti, aby neovplyvňoval priebeh levitácie. Takýmito materiálmi sú diamagnety, čo je väčšina plastov, prípadne hliník, ktorý je sice paramagnetický ale magnetické pole naň vpýva veľmi slabo.

### 3.2.3 Priestor pre levitáciu

V Kapitole 1.1 som uviedol základnú podmienku levitácie, a to že veľkosť nadnášavej sily  $\vec{F}_{vz}$  musí byť rovnaká ako veľkosť gravitačnej sily  $\vec{F}_G$ . Tieto dve sily nie sú však jedinými silami, ktoré na levitujúci magnet pôsobia. Odpór prostredia a prúdenie vzduchu vytvárajú sice veľmi malé sily v rôznych smeroch, ale vzhľadom na malú hmotnosť magnetu, sú schopné ho pri levitácii rozkmytať. Toto kmitanie je pre nás vysoko nežiadúce.

Problém spočíva v tom, že pri kmitaní magnetu zo strany na stranu sa pól magnetu vychyľuje tak, že sa nenachádza priamo oproti hallovemu senzoru. Kedže magnetické pole magnetu je silne nehomogénne, dochádza tým aj k poklesu vzniknutého napäťa na hallovom senzore, čo vytvára nepravdivú informáciu o polohe magnetu voči senzoru<sup>3</sup>. Toto kmitanie má za následok rýchle zmeny prúdu, bez ohľadu na skutočnú výšku magnetu, čo viedie k ďalšiemu rozkmitávaniu. Kmitanie môže časom dosiahnuť až takú veľkosť, že viedie k nestabilite systému a následnému vyleteniu magetu.

Riešením je ohraničenie a uzavretie levitujúceho predmetu, ktoré ho odizoluje od vplyvov okolitého prostredia. Toto ohraničenie som realizoval pomocou prieľadnej trubky s vnútorným priemerom 10 mm a vonkajším priemerom 12 mm (levitujúci magnet je v tvare disku o priemere 8 mm). Trubka má dostatočné rozmer, aby nebránila magnetu v pohybe, no zároveň môže slúžiť aj na stabilizovanie procesu v prípade, ak sa začne z nejakého dôvodu magnet rozkmitávať.

Ako som naznačil v Kap. 1.4, môže nastať situácia, kedy sa magnet sám pritiahnne k oceľovému obalu elektromagnetu. Vzhľadom na to, že levitácia bude prebiehať v priestore ohraničenom priesvitnou trubkou, riešením je trubku zhora uzavrieť v takej vzdialosti, v ktorej ešte neplatí pre sily medzi obalom a magnetom  $|F_m| \geq |F_G|$ . To sa dosiahne pomocou plastového veka nalepeného zhora. Experimentálne som zistil, že najväčšia vzdialenosť, pre ktorú výraz ešte neplatí je 9 mm (viď Obr. 3.3). Plastová trubka je žiaľ, rovnako ako hliníkový profil, tovar predávajúci sa v celku vo väčšej dĺžke, takže je nutné pred namontovaním ešte dodatočne spraviť dĺžkovú úpravu.

Kedže Magneto bude využívaný v prvom rade ako výukový prostriedok, je dôležité, aby sa v prípade nesprávneho riadiaceho programu, čoho dôsledkom môže byť otočenie magnetu, dala priesvitná trubka ľahko odmontovať a bolo možné sa k magnetu dostať. Ak sa totiž magnet v dôsledku zlého riadenia otočí, dôjde k výmene pólov a elektromagnet bude magnet namiesto príťahovania odpudzovať.

Tento jednoduchý a rýchly prístup k magnetu som vyriešil pomocou gumeného tesniaceho prstanca, ktorý sa na PCB nalepí. Trubka sa doň bude zasúvať. Jeho vnútorný priemer je 11,8 mm, čo je trošku menší ako je vonkajší priemer trubky a jeho výška sú 2 mm, takže trubka v ňom bude držať a v prípade potreby je možná jej rýchla demontáž.

## 3.3 Zhotovenie levitátora Magneto

Po návrhu schémy, voľbe správnych komponentov, a vytvorení PCB je potrebné celý návrh zrealizovať a z komponentov spolu s PCB vytvoriť funkčný celok.

V predchádzajúcim texte som naznačil zopár dôležitých faktov, týkajúcich sa konštrukovania levitátora Magneto. V tejto časti v skratke zhrniem a priblížim tie najdôležitejšie.

<sup>3</sup>Dostávame informáciu, že magnet sa nachádza ďalej od senzora - v prípade Magneta vyššie a treba zmenšiť veľkosť elektrického prúdu, čo má za následok, že magnet začne predčasne padat.



Obr. 3.4: Kompletný levitátor Magneto spojený s Arduinom UNO.

Skoro všetky komponenty sa pripájajú k PCB procesom nazývaným spájkovanie. Ide o metalurgické spojovanie predmetov roztavenou spájkou, v našom prípade cínom. Kedže pri spájkovaní sa dosahujú teploty až  $350\text{ }^{\circ}\text{C}$  je veľmi dôležité dbať na bezpečnosť a voliť také poradie prispájkovania predmetov, aby dotyčný mal vždy dostatok priestoru na manipuláciu a nedošlo k zbytočnému poraneniu alebo znehodnoteniu nejakej súčiastky.

Pri spájkovaní vzniká trvalý spoj, ktorý nie je určený na viacnásobné odstraňovanie a opäťovné vytváranie. V prípade potreby je možné spoj odstrániť, ale pokiaľ možno, je dobré sa takýmto situáciám vyvarovať, lebo ako PCB, tak aj súčiastka sú vystavované pri odpájkovaní extrémnym podienkam a môžu sa poškodiť. Problematicosť odstránenia spoja ešte zvyšuje fakt, že skoro všetky použité komponenty sú v SMD balení a manipulácia s malými predmetmi pri vysokej teplote je pomerne riskantná.

Z týchto dôvodov sa treba pred trvalým pripojením súčiastky k PCB uistíť, že jej orientácia je správna a každý z jej pinov naozaj pripájame na miesto, ktoré je preň vytvorené.

Pri pripájaní elektromagnetu odporúčam, vopred si odskúšať, ktorý z jeho káblov má byť pripojený ku kladnému a ktorý k zápornému znamienku (na PCB označené “+” a “-” vid’ Obr. 3.1). Síce levitujúci neodimový magnet má dva póly a teoreticky by nemalo záležať na polarite elektromagnetu, ale jedna zo strán magnetu je pôsobením zemského magnetického poľa ľahšia k zemi silnejšie ako druhá. V prípade, že na magnet nepôsobí elektromagnetom generované magnetické pole, má snahu sa natočiť podľa zemského magnetického poľa. To môže spôsobovať rozkmitávanie počas levitácie, prípadne celú levitáciu znemožniť.

Správnu polaritu elektromagnetu zistíme tak, že najprv určíme, ktorá strana magnetu je ľahšia k zemi silnejšie - keď disk postavíme na kánt vždy sa zvalí na jednu stranu, následne elektromagnet pripojíme k zdroju napäťia a ak je takto položený disk elektromagnetom pritiahovaný, polatira zapojenia je správna. V opačnom prípade treba otočiť zapojenie elektromagnetu.

Veľmi praktickou pomôckou je označenie severného a južného pólu magnetu, aby bolo vždy zrejmé, akým spôsobom sa do levitátora má vkladať a v prípade nepodareného pokusu nie je potreba nanovo určovať jeho polaritu.

# 4 Softvérové rozhranie levitátora Magneto

Elektromagnetický levitátor Magneto je jednou zo súčasťí školského projektu AutomationShield [3]. Zmyslom tohto projektu je vytvorenie skupiny shieldov kompatibilných s prototypizačnými doskami Arduino. Tieto shieldy budú slúžiť ako výukové prostriedky, na ktorých si študenti môžu precvičiť aplikáciu nadobudnutých vedomostí na reálnych sústavách, predovšetkým riadiace algoritmy, akým je napríklad PID regulátor (čo je to PID regulátor priblížim v Kap. 5.1).

Každý zo shieldov funguje na inej fyzikálnej báze a spracúva iný druh informácií, avšak bolo by veľmi neefektívne, keby používateľ musel zakaždým študovať spôsob obsluhy a fungovania každého zo shieldov. Z tohto dôvodu je snaha ovládanie všetkých shieldov pokiaľ možno čo najviac unifikovať, prípadne vytvoriť súbor podobných funkcií typických pre všetky zariadenia, ktoré budú používateľovi prácu so shieldom zjednodušovať. K tomuto cielu sa v rámci projektu AutomationShield vytvorila jednotná knižnica "AutomationShield" obsahujúca niekoľko začlenených súborov, popisujúcich funkcie všeobecné pre všetky shieldy (napríklad ovládanie PID regulátora a vzorkovanie), ako aj funkcie typické pre obsluhu konkrétnej dosky (napr. funkcie na obsluhu levitátora Magneto).

## 4.1 Knižnica

Pri tvorbe programu sa často krát stretáme s problémom, že potrebujeme napísať niekoľko riadkov, ktoré sme už raz v programe uviedli. Riešime to kopírovaním predoších častí programu. Keď však musíme kopírovanie tej istej časti zopakovať 10-krát, hľadáme spôsob ako sa tomu vyhnúť. Riešením je napísať v programe funkciu a miesto kopírovania celej 20 riadkovej state kopírjeme len jeden riadok s názvom funkcie. Ak ale pracujeme na viaceru programoch a v každom využívame nejaké funkcie, ktoré sa opakujú, snažiť sa ich zakaždým nájsť v iných programoch a skopírovať ich, je veľmi nepraktické riešenie. O to viac, ak sú tieto programy určené pre ďalších používateľov, ktorí s nimi budú pracovať a snažiť sa ich ďalej rozvíjať. Práve z tohto dôvodu vznikli knižnice. Ide o súbory, ktoré možno veľmi jednoducho začleniť do programu, obsahujúce preddefinované funkcie opisujúce zvyčajne komunikačné protokoly alebo základné operácie vzťahujúce sa k objektu, pre ktorý boli napísané. Ich úlohou je programátorovi zjednodušiť prístup k využívanému rozhraniu a zefektívniť jeho prácu s ním. Mali by byť vytvorené tak, aby programovanie zefektívňovali, no nijakým spôsobom neobmedzovali.

Pre ovládanie shieldu Magneto bola taktiež vytvorená knižnica. Zjednodušuje komunikáciu medzi používateľom a jednotlivými komponentami shieldu a ponúka niekoľko užitočných funkcií, ktoré sú potrebné v procese levitácie.

Každá knižnica pozostáva z minimálne dvoch súborov a to *header* a *source*.

#### 4.1.1 Header

Ako názov naznačuje, header je hlavičkou nami vytváratej knižnice. Ako som spomenul, knižnica slúži na to, aby sme nemuseli zakaždým definovať funkcie, ktoré sme už veľa krát použili a ktoré ešte veľa krát použijeme. V súbre header deklarujeme všetky funkcie a premenné, ktoré chceme do knižnice začleniť. Zároveň sa tento typ súboru používa pri vkladaní knižnice do programu (súbor *source* sa nikdy nevkladá do iných súborov).

Na začiatku, pri tvorbe *header* súboru deklarujeme triedu. Trieda je súhrn rôznych funkcií, ktoré majú minimálne jednu rovnakú vlastnosť a tou je práve trieda. Inými slovami, trieda je spôsob akým dávame dokopy rôzne funkcie a premenné a vysvetľujeme im, že vzájomne súvisia. Vďaka tomu, že sa nachádzajú v jednej triede, môžme dané funkcie vzájomne kombinovať a prepájať a na základe toho vytvárať ešte komplexnejšie celky, ktoré budú dostupné koncovému užívateľovi [7]. V rámci jedného *header* súboru možno deklarovať ľubovoľný počet tried podľa potreby<sup>1</sup>.

V prípade mnou napísanej časti knižnice som deklaroval triedu nazvanú *MagnetoShieldClass*. Za názvom triedy nasleduje v zložených zátvorkách zoznam všetkých funkcií a premenných, ktoré budú v programe použité. Pri ich deklarácií je dôležité uviesť o aký typ premennej sa jedná, v prípade funkcie či má návratovú hodnotu alebo vstupné parametre a ak áno, akého dátového typu sú:

```
public:  
    void begin();                                //funkcia bez navratovej hodnoty  
    uint8_t setVoltageV(float voltageV); //vstupny parameter float,  
                                         //navratova hodnota typu uint8_t  
    float readHeight();                         //navratova hodnota typu float  
private:  
    float Min;                                  //premenna typu float  
    float Max;                                 //premenna typu float
```

V rámci triedy si možno všimnúť, že funkcie a premenné píšem do dvoch skupín a to *private* a *public*. Toto rozdelenie funkcií je zrejmé aj z názvu samotných častí.

V časti *private* deklarujem funkcie, ktoré sú využívané v rámci triedy a používam ich pri vytváraní ďalších funkcií. Ak by som sa rozhadol vytvoriť inú triedu spolupracujúcu s už napísanou, k týmto funkciám nemám prístup a kompilátor (prekladač kódu) ich nerozozná. Rovnako to funguje aj pri využívaní celej knižnice. K funkciám a premenným v tejto časti užívateľ nemá prístup a nemôže s nimi pracovať.

Časť *public* znova obsahuje funkcie, ktoré môžem používať, jednak pri tvorbe iných funkcií v rámci triedy, jednak pri tvorbe iných tried. Tieto funkcie budú dostupné aj iným *header* súborom tvoriacim knižnicu a aj koncovému užívateľovi využívajúcemu danú knižnicu.

Za deklaráciou premenných a uzavretím triedy ďalšou kučeravou zátvorkou sa nachádza jednoriadkový kód:

```
extern MagnetoShieldClass MagnetoShield;
```

Tento zápis hovorí o tom, pod akým názvom bude trieda *MagnetoShieldClass* vystupovať v súboroch, v ktorých budem triedu využívať. Akonáhle sa rozhodnem v hlavnom programe zavolať nejakú funkciu z tredy, budem ju volať pomocou názvu *MagnetoShield* s bodkovou konvenciou a názvom samotnej funkcie napr.:

```
MagnetoShield.calibration();
```

<sup>1</sup>V mojom pípade stačí len jedna trieda. Iné triedy, ktoré budú taktiež obsahom vytváratej knižnice na obsluhu *shieldov*, budú deklarované v samostatných *header* súboroch.

Ak chceme napísaný *header* súbor využívať v rámci iných programov, musíme ho zakaždým do toho programu pričleniť pomocou príkazu `#include "NazovSuboru.h"`. Problém by nastal v prípade, že nejakým nedopatrením by sme *header* pridali do programu viac krát. Počítač by videl dve rozdielne triedy, ktoré sa chovajú rovnako a to by viedlo ku chybovému hláseniu. Pred takouto situáciou sa chránime pridaním podmienky na začiatok *header* súboru, ktorá kontroluje, či už do programu náš *header* bol pridaný alebo nie. V prípade, ak áno, komplilátor celú deklaráciu tried aj s funkciemi preskočí a pokračuje v komplilácií programu[15]. Nižšie je uvedený príklad ochrany pre mnou vytvorený súbor "MagnetoShield.h", ktorý je súčasťou knižnice AutomationShield.

```
#ifndef MagnetoShield_h
#define MagnetoShield_h
...sem ide obsah header
#endif
```

Ďalšou časťou *header* súboru, v mojom prípade "MagnetoShield.h", je začlenenie iných využívaných *header* súborov, s ktorými funkcie v rámci triedy pracujú.

Kedže ovládanie DA prevodníka PCF8591T je realizované pomocou komunikačného protokolu I<sup>2</sup>C, je potrebné túto komunikáciu zadefinovať. Vzhľadom na to, že opäťovné naprogramovanie protokolu I<sup>2</sup>C by bola veľmi zdlhavá a komplexná úloha, využil som na to už vopred spravenú knižnicu s názvom "Wire" [2].

Druhý súbor ktorý som, pridal do mnou vytváratej časti knižnice je "AutomationShield.h". Začlenením tohto súboru do môjho *header* som schopný používať všetky *public* funkcie deklarované vo všetkých triedach "AutomationShield.h". Zároveň sa tým stáva môj *header* súbor "MagnetoShield.h" súčasťou súboru tried deklarovaných v "AutomationShield.h". Je to spôsob, akým sa dá pospájať viaceru knižnic, obsluhujúcich množstvo zariadení do jednej knižnice.

Oba súbory sa pridajú pomocou príkazu `#include "NazovSuboru.h"` na začiatku *header*, hned' za ochrannú podmienku proti viacnásobnému začleneniu.

Posledná čas kódu, ktorú som ešte neopísal je príkaz `#define PCF8591 (0x90 >> 1)` a `#define HallSenzor A3`. Ide o makro definíciu nejakej konštanty, platnú pre celý kód. Príkaz vyhľadá všetky názvy, ktoré sú rovnaké ako prvý názov za `#define` a nahradí ich tým, čo sa nachádza za ním. V našom prípade sa všetky názvy PCF8591 nahradia 8-bitovou adresou v hexadecimálnom tvare posunutou o jedno miesto do prava (definuje sa adresa DA prevodníka pre I<sup>2</sup>C komunikáciu) a všetky názvy HallSenzor sa nahradia zápisom A3, čo určuje pin vrámcu dosky Arduino UNO.

Kompletný kód *header* súboru "MagnetoShield.h" nájdete v prílohe A.

#### 4.1.2 Source

*Source* predstavuje zdrojový súbor pre knižnicu (od toho aj názov source z ang. zdroj). Ako tomu bolo v prípade *header*, knižnica môže obsahovať aj viacero *source* súborov. V predchádzajúcom súbore sa premenné a funkcie delkarovali, ale ako si môžeme všimnúť, nikde nie je napísané, čo ktorá funkcia má robiť, prípadne aké hodnoty majú premenné nadobúdať. Presne tieto veci rieši súbor *source*.

Najprv je potrebné poukázať na to, akým funkciám ideme pridelovať vlastnosti. Z toho dôvodu sa na začiatku dokumentu nachádza príkaz `#include "MagnetoShield.h"`, pomocou ktorého vytvoríme prepojenie medzi opísaným *header* súborom "MagnetoShield.h" a vytváraným dokumentom definujúcim jednotlivé prvky. Dalej súbor obsahuje už len sa-

motné funkcie a vzťahy medzi nimi. Kedže sa v jednom *header* súbore môže nachádzať viacero tried, ako prvé vždy zadefinujeme, v ktorej triede, ktorej funkcie ideme pridelovať vlastnosti. Za označením funkcie sa v kučeravých zátvorkách nachádzajú príkazy.

Ďalej priblížim jednotlivé funkcie, ktoré som napísal pre obsluhu levitátora Magneto<sup>2</sup>.

Prvou z funkcií je inicializačná funkcia. Pomocou nej, ako som už spomíнал, sa definuje pin A3 na doske Arduino UNO, ako vstup pre analógový signál a zároveň sa ňou spúšťa knižnica "Wire.h" zodpovedná za I<sup>2</sup>C komunikáciu. Využitie funkcie `Wire.begin()` vrámci tvorby inej knižnice je presne príklad toho, ako sa môžu kombinovať funkcie z viacerých knižníc, pri tvorbe novej.

```
void MagnetoShieldClass::begin()
{
pinMode(HallSenzor, INPUT); //nastavi HallSenzor (pin) ako INPUT (vstup)
Wire.begin(); //inicIALIZUje "Wire" kniznicu
}
```

Ďalšia funkcia je zodpovedná za vytváranie požadovaného napäťa. Ide o funkciu, ktorá má jeden vstupný parameter (argument) v tvare čísla *float* (číslo s desatinou čiarkou) a má aj návratovú hodnotu. Tým sa myslí, že po prebehnutí funkcie, funkcia vráti nejakú hodnotu, ktorá môže byť pridelená premennej. Návratová hodnota tejto funkcie má rozmer 8-bitového čísla a v dekadickom tvaru môže nadobúdať hodnoty 0-255.

Úloho funkcie je na základe argumentu ktorý môže nadobudnúť hodnotu 0.00-5.00 vytvoriť požadovanú veľkosť napäťa. Funkcia musí tento vstupný interval premeniť na 8-bitové číslo a následne ho pošle DA prevodníku, ktorý vytvorí požadované napätie 0-5 V.

Ako je zrejmé z rozmerou vstupného argumentu, táto funkcia je napísaná skôr na obsluhu DA prevodníka a prípadnú experimentálnu prácu s Magnetom. Pre zasielanie vstupných informácií do DA prevodníka v rámci regulačného procesu je vhodnejšia funkcia `MagnetoShield.setVoltage(float u)` uvedená nižšie.

Komunikácia s DA prevodníkom prebieha v štyroch krokoch. Ako prvé sa zariadenie osloví pomocou adresy, ktorú má predpísanú od výroby. Následne sa pošle kontrolný bajt, ktorý definuje akú operáciu od PCF8591T vyžadujeme (slúži ako DA ale aj AD prevodník). Ďalší bajt hovorí o hodnote generovaného napäťa. Posledná správa posielaná na PCF je ukončenie komunikácie. Ukončenie komunikácie neznamená ukončenie činnosti DA prevodníku, len oznamuje čipu, že nemá očakávať žiadne ďalšie dátá (pre podrobnejší popis priebehu komunikácie viď datasheet [11]).

```
uint8_t MagnetoShieldClass::setVoltageV(float voltage)
{
    if(voltage > 5.00){ //ohranicenie intervalu 0.00-5.00
        voltage = 5.00;
    } else if (voltage < 0.00){
        voltage = 0.00;
    }
    float voltageF = voltage*255.00/5.00; //konverzia intervalu na
    //0.00-255.00
    uint8_t voltageB = (uint8_t) voltageF; //konverzia float na 8-bit
    Wire.beginTransmission(PCF8591); //oslovenie PCF cez adresu
    Wire.write(0x40); //zvolenie funkcie DA prevodniku
    Wire.write(voltageB); //hodnota generovaneho napatia
```

<sup>2</sup>Prototypizačné dosky Arduino sa programujú pomocou programovacieho jazyka C/C++, ktorý však využíva už vopred urobené knižnice, ktoré prístup k celej doske a jej funkciám zjednodušujú. Tieto knižnice sú začlenené do obsahu knižnice AutomationShield ale ozrejmovať činnosť ich funkcií v nasledujúcich častiach nebudem

```

        Wire.endTransmission();           //ukoncenie komunikacie
        return voltageB;                //navratova hodnota funkcie
    }

```

Asi najdôležitejšou informáciou pre levitátor Magneto je určenie hraníc, v ktorých má predmet lietať. Ide o to, že Magneto na určovanie polohy magnetu používa hallov senzor. Ten je schopný relatívne presne určiť zmenu polohy magnetu, ale výstupná informácia z neho nie je konkrétna poloha magnetu, ale len hodnota magnetickej indukcie ktorú nameenal (výstupom je napäťie zodpovedajúce určitej indukcii). Pred procesom levitácie treba nejakým spôsobom zadefinovať, aká hodnota magnetickej indukcie zodpovedá najvyššej možnej polohe, ktorú môže magnet dosiahnuť a aká hodnota predstavuje najnižšiu polohu. Za určovanie týchto hraníc zodpovedá kalibračná funkcia.

Predpokladáme, že levitácia magnetu bude začínať zo zeme. Hallov senzor zmeria magnetickú indukciu a priradí jej hodnotu privátnej premennej *Min*. V tomto momente DA prevodník nevytvára žiadne napätie a elektromagnet je vypnutý. Následne Arduino pošle príkaz DA prevodníku v podobe 8-bitového čísla a prevodník vytvorí maximálne napätie. Tým sa MOSFET plne ovorí a elektromagnetom tečie maximálny prúd. Zapnutý elektromagnet pritiahe magnet do vrchnej polohy (magnet sa neprisaje k elektromagnetu vďaka trubici, ktorá ohraničuje jeho pohyb, viď 3.2.3). Následne hallov senzor opäť zmeria hodnotu indukcie a priradí ju privátnej premennej *Max*. Po nakalibrovaní (určení hraníc) sa DA prvdník opäťovne vypne, aby levitácia začala z dolnej polohy.

Spracovanou informáciou z hallovho senzoru je 10-bitové dekadické číslo (nadobúda hodnoty 0-1023)<sup>3</sup>. Premenné *Min* a *Max* sú však deklarované ako desatiné, tým pádom aj nameraná hodnota sa do nich uloží v tvare s desatinou čiarkou. Tento tvar je potrebný pri spracovávaní hraníc inými funkciami.

```

void MagnetoShieldClass::calibration()
{
    Wire.beginTransmission(PCF8591); //I2C komunikacia
    Wire.write(0x40);
    Wire.write(0);
    Wire.endTransmission();
    delay(1000);                  //caka na vytvorenie podmienok
    Min = analogRead(HallSenzor); //najnizsia poloha magnetu

    Wire.beginTransmission(PCF8591); //I2C komunikacia
    Wire.write(0x40);
    Wire.write(255);
    Wire.endTransmission();
    delay(1000);                  //caka na vytvorenie podmienok
    Max = analogRead(HallSenzor); //najvyssia poloha magnetu

    Wire.beginTransmission(PCF8591); //nastavenie vzletu na zem
    Wire.write(0x40);
    Wire.write(0);
    Wire.endTransmission();
    delay(1000);
}

```

Akonáhle vykonáme kalibráciu, zistené hodnoty sa uložia do globálnych premenných, ale užívateľ nie je schopný dané hodnoty iba za pomocí kalibračnej funkcie využívať. Pre tento účel som vytvoril dve jednoduché funkcie, ktoré vracajú hodnotu maxima *Max* a

<sup>3</sup> Je dôležité pripomenúť, že hallov senzor, ktorý sa nachádza na Magnete je bipolárny a stavu bez magnetického poľa zodpovedá hodnota približne 512. Maximum a minimum sa bude kvôli zvolenej polarite magnetu a elektromagnetu na Magnete pohybovať v intervale hodnôt 0-512

minima *Min*. Kedže tieto hodnoty sú uložené v premennej ako desatiné číslo, funkcia priamo vykoná konverziu na celočíselnú hodnotu *integer* (skratka int).

```
int MagnetoShieldClass::getMin()
{
    int variable = (int) Min;      //konverzia float->int
    return variable;
}

int MagnetoShieldClass::getMax()
{
    int variable = (int) Max;      //konverzia float->int
    return variable;
}
```

Ako som uviedol pri opise funkcie kalibrácie, ale aj na mnohých iných miestach vrámci tejto práce, informácia o aktuálnej polohe levitujúceho predmetu je kľúčová pre stabilné udržanie tohto procesu a na jeho správnu reguláciu. Preto je potrebné definovať funkciu podávajúcú informáciu o polohe magnetu v rámci intervalu hodnôt maxima a minima.

Z rovnakého dôvodu ako tomu bolo pri deklarácií premenných *Max* a *Min*, aj táto funkcia vracia celé číslo získané z hallovho senzoru ako číslo s desatinou čiarkou. Ale tým že hodnota má vždy na pozícii desatiných čísel nuly, v prípade potreby jej následná konverzia na celočíslo nie je problémová.

```
float MagnetoShieldClass::readHeight()
{
    float Height = analogRead(HallSenzor); //momentalna poloha lietania
    return Height;
}
```

Levitátor Magneto je v prvom rade určný na precvičovanie riadiacich algoritmov v snahe dosiahnuť plinulú a ustálenú levitáciu. Za týmto učelom som vytvoril ďalšu z funkcií, pomocou ktorej jednoducho definujeme výšku v akej má vznášanie sa prebiehať. Táto funkcia má jeden vstupný argument typu *float*, do ktorého sa zadáva výška levitácie v percentách. Pre tento spôsob zadávania výšky som sa rozhodol z dôvodu, že na začiatku každého procesu lietania sa zvičajne realizuje kalibrácia. Výstupom kalibrácie sú zvičanie hodnoty väčšie rôzne od nuly (magnet určený na levitáciu nie je schopný vytvoriť tak silné pole, aby hallov senzor vrácal hodnotu 0), tým pádom výpočet výšky v ktorej chceme levitáciu realizovať, by bol zbitočne zložitý.

Táto funkcia využíva vo vnútri inú funkciu zo spoločnej knižnice *AutoamtionShiel* a je volaná ako *AutomationShield.mapFloat(set,0.00,100.00,Min,Max)*. Táto funkcia prepočítava argument zadaný v percentách na číselnú hodnotu v intervale ohraničenom globálnymi premennými *Min* a *Max*.

```
float MagnetoShieldClass::setHeight(float set)
{
setpoint = AutomationShield.mapFloat(set,0.00,100.00,Min,Max);
return setpoint;
}
```

Funkciu *MagnetoShield.setVoltage(float u)* som napísal pre potreby tvorby regulačných programov, ako je napríklad regulácia pomocou PID. Funkcia je veľmi stručná, bez návratovej hodnoty a ako argument má desatiné číslo, ktoré je následne prekonvertované do 8-bitovej podoby. Pomocou neho je ovládané napätie na DA prevodníku.

```
void MagnetoShieldClass::setVoltage(float u)
{
```

```

    uint8_t u8B = (uint8_t) u;           //konverzia float->uint8_u
    Wire.beginTransmission(PCF8591);
    Wire.write(0x40);
    Wire.write(u8B);
    Wire.endTransmission();
}

```

Posledná mnou napísaná funkcia pre ovládanie levitátora Magneto priamo súvisí s regulačným algoritmom PID. Pomocou tejto funkcie sa vykoná rozdiel medzi nami zadanou požadovanou hodnotou lietania definovanou v `MagnetoShield.setHeight(float set)`, ktorá sa uložila do globálnej premennej *setpoint* a aktuálnou polohou magnetu zmeranou hallovým senzorom. Výstupom z funkcie je takzvaná regulačná odchýlka.

```

float MagnetoShieldClass::error()
{
    float position = analogRead(HallSenzor); //momentalna pozicia magnetu
    float err = setpoint - position;          //setpoint z funkcie setHeight
    ()
    return err;
}

```

Kompletný kód *source* súboru “MagnetoShield.cpp” nájdete v prílohe B.

# 5 PID

V predchádzajúcich častiach som vysvetlil, ako som elektromagnetický levitátor Magneto navrhol, zostrojil a vytvoril preň užívateľské rozhranie v podobe dvoch súborov, ktoré sú súčasťou knižnice pre ovládanie všetkých shieldov v rámci projektu AutomationShield. V tejto časti by som sa rád venoval ukážke praktického príkladu a princípom použitých v ňom. Nasledujúcu kapitolu možno považovať za ukážku, ako sa dá pracovať s levitátorom Magneto.

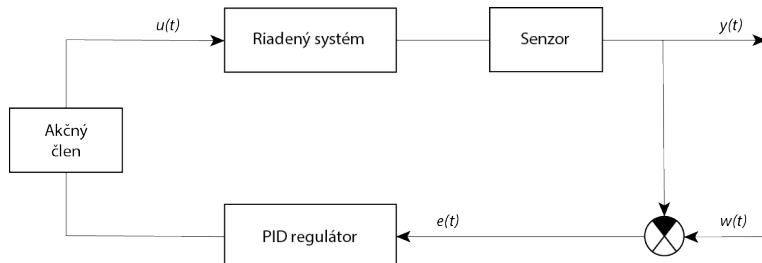
## 5.1 Spätnoväzbové riadenie - PID regulátor

Predtým, ako uvediem príklad programu pre riadenie PID regulátorom, by som rád priblížil čo pojmy spätnoväzbové riadenie a PID regulátor znamenajú.

Spätnoväzbové riadenie, je riadenie ktoré ovplyvňuje sústavu na základe získaných informácií o aktuálnom stave, v ktorom sa sústava nachádza. V princípe, riadiaca jednotka zosníma regulovanú veličinu, porovná ju s požadovanou hodnotou a na základe toho sa rozhodne ako ovplyvní sústavu. Príkladom môže byť tempomat na automobile. Zariadenie zodpovedné za udržanie rýchlosťi pravidelne sníma, akou rýchlosťou sa automobil pohybuje, zistenú rýchlosť porovná s požadovanou hodnotou zadanou vodičom a na základe toho sa rozhodne, či pribrzdi alebo pridá.

Na realizáciu spätnoväzobného riadenia existuje množstvo rôznych algoritmov, ktoré sú zodpovedné za to, ako sa riadiaca jednotka rozhoduje. Samozrejme, čím prepracovanejší spôsob "rozmýšľania", tým sú kladené aj vyššie požiadavky na riadiacu jednotku zodpovednú za výpočty. V mnohých prípadoch je potrebné regulovanú veličinu zosnímať a vyhodnotiť veľa krát za sekundu (môžu to byť aj tisícky krát). To v kombinácii so zložitým regulačným algoritmom môže klásiť až nesplnitelné požiadavky na riadiacu jednotku (riadiaci počítač). Preto je nutné pri návrhu riadiaceho systému zvážiť, aký regulačný algoritmus sa použije a či požiadavky na riadiacu jednotku sú splnitelné.

Teraz sa dostávame k tomu, čo je to PID regulátor. Je to spôsob, akým sú spracovávané zosnímané údaje zo sústavy - výstupy  $y(t)$  a na základe nich vytvorené akčné zásahy



Obr. 5.1: Schéma riadenia PID regulátorom

do sústavy - vstupy  $u(t)$  (veličina upravujúca regulovaný systém), teda je to typ spätnoväzbového riadenia (viď Obr. 5.1). Vstupy do sústavy sú vypočítané na základe rozdielu požadovanej hodnoty  $w(t)$  a zosnímaného výstupu  $y(t)$ . Tento rozdiel sa tiež nazýva regulačná odchýlka  $e(t)$ . Výhodou tohto algoritmu je jeho pomerná jednoduchosť, tým pádom nekladie vysoké požiadavky na výpočtový systém.

Skratka PID predstavuje tri zložky, z ktorých sa regulátor skladá:

- P zložka - ide o takzvanú proporcionálnu časť regulátora. Vstup do sústavy  $u(t)$  je priamo úmerný veľkosti regulačnej odchýlky  $e(t)$ . Tá je násobená takzvanou proporcionálnou konštantou  $K_p$ , ktorá je nastaviteľným parametrom. Výsledkom násobenia je požadovaný vstup (Rov.(5.1)).

Nesprávna voľba konštanty  $K_p$  môže viesť k nedostačujúcej regulácii. Ak  $K_p$  je príliš malá, systém sa k požadovanej hodnote blíži veľmi pomaly. Naopak, ak je veľká, vedie to k rozkmitávaniu sústavy a v prípade príliš veľkej proporcionálnej konštanty sa systém dostáva za hranicu stability a jeho regulácia nie je možná. Hodnota vstupu získaná proporcionálnou zložkou je:

$$u(t) = K_p e(t) \quad (5.1)$$

- I zložka - predstavuje integračnú zložku regulátora. Integrál regulačnej odchýlky cez celú dobu regulácie hromadí hodnoty týchto odchýlok, sčítava ich a na základe toho vytvára vstupy. Ak je momentálna regulovaná veličina menšia ako požadovaná hodnota, regulačná odchýlka je kladná a integrálna časť PID narasá. Ak je hodnota reguloanej veličiny väčšia, hodnota integrálu začína klesať.

Integrál je následne násobený integračnou konštantou  $K_i$ . Príliš veľká hodnota  $K_i$  spôsobuje kmitanie okolo požadovanej hodnoty regulácie.

Veľkosť akčného zásahu integrálnej časti PID teda závisí jednak od veľkosti regulačnej odchýlky, jednak od doby, ako dlho sa regulačná odchýlka v systéme nachádza a od integračnej konštanty  $K_i$ . Následne integračná zložka generuje vstupné hodnoty:

$$u(t) = K_i \int_0^t e(\tau) d\tau \quad (5.2)$$

- D zložka - alebo tiež derivačná zložka PID. Pomocou derivácie regulačnej odchýlky v čase berie do úvahy, s akou rýchlosťou sa odchýlka mení a na základe toho vytvára vstup do sústavy. Snaží sa predchádzať ďalším vznikom regulačných odchýliek.

Derivácia je násobená derivačnou konštantou  $K_d$ . V prípade veľkej hodnoty derivačnej konštanty je systém stabilnejší, lebo derivačná zložka mu bráni aby nadobúdal iné hodnoty ako požadovanú. Avšak v tomto prípade dosiahnutie požadovanej hodnoty trvá dlhšie, kedže derivačná zložka pôsobí voči zmene systému:

$$u(t) = K_d \frac{de(t)}{dt} \quad (5.3)$$

V Tab. 5.1 je krátke zhrnutie, akým spôsobom konštanty PID regulátora ovplyvňujú priebeh regulácie [10].

Tabuľka 5.1: Vplyv konštánt na priebeh regulácie

| Konštantá | Čas rastu | Prekmitnutie | Čas ustálenia    |
|-----------|-----------|--------------|------------------|
| $K_p$     | Znižuje   | Zvyšuje      | Nemá veľký dopad |
| $K_i$     | Znižuje   | Zvyšuje      | Zvyšuje          |
| $K_d$     | Zvyšuje   | Znižuje      | Znižuje          |

Akonáhle spojíme činnosť všetkých troch častí dohromady, vznikne PID regulátor, ktorého výstup je ovplyvňovaný všetkými tromi zložkami P, I, D a opisuje ho vzťah:

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de(t)}{dt} \quad (5.4)$$

Vzťah (5.4) definuje činnosť regulátora v časovej oblasti. Vzhľadom na jeho široké uplatnenie vzniklo aj množstvo iných foriem a variacií, ktorými možno daný algoritmus zapísat - napríklad zápis PID v obrazovej oblasti. Pre potreby využitia regulátora v levitátore Magneto je pre nás zaujímavá forma nazývaná číslicový PID regulátor. Ide o transformáciu Rov. (5.4) do diskrétnej podoby [4]

$$u(kT) = K_p \{e(kT) + \frac{T}{T_i} \sum_{i=0}^k e(iT) + \frac{T_d}{T} [e(kT) - e[(k-1)T]]\} \quad (5.5)$$

pričom vzťahy medzi  $K_i$ ,  $K_d$  a  $T_i$  - integračná časová konštanta,  $T_d$  - derivačná časová konštanta sú:

$$T_i = \frac{K_p}{K_i} \quad (5.6)$$

$$T_d = \frac{K_d}{K_p} \quad (5.7)$$

Diskrétna forma PID je vhodná na použitie v číslicových regulátoroch, ináč povedané v počítačoch. Počítač nie je schopný merať veličiny nepretržito. Miesto toho dátá získava v pravidelných intervaloch, diskrétnych okamihoch  $kT$ , kedy zmeria okamžitú hodnotu sledovanej veličiny. Frekvencia s akou získava dátu sa nazýva vzorkovanie. V Rov. (5.5) je periódna vzorkovania označená ako  $T$  a rádovo nadobúda hodnoty od stoviek mikrosekúnd až po desiatky sekúnd.

Práve tento spôsob interpretácie PID regulátora je využívaný aj v knižnici AutomationShield a je súčasťou inštruktážného príkladu pre levitátor Magneto.

## 5.2 Príklad použitia PID v levitátore Magneto

Magneto je síce zariadenie navrhnuté tak, aby bolo schopné zdvihnúť a udržať v priestore špecifický typ magnetu, ale bez logicky zostaveného riadiaceho programu, nedokáže informácie správne spracovať a vytvárať na ich základe zmysluplné vstupy. Kedže Magneto je shield určený pre prototypizačnú dosku Arduino UNO, ktorá ho ovláda, Arduino predstavuje riadiacu jednotku celého levitátora. Zhromažďuje namerané informácie a posielá dátu, na základe ktorých DA prevodník ovplyvňuje levitačnú sústavu.

Program, pre ktorý som nasledujúci opis urobil, sa nachádza v prílohe C.

Na začiatku tvorby programu som pridal knižnicu `#include <AutomationShield.h>`<sup>1</sup>, ktorá obsahuje všetky používateľské funkcie na obsluhu shieldu Magneto a s tým spojených operácií, ako zadávanie parametrov PID, nastavenie vzorkovania, atď.

Na začiatku programu deklarujeme globálne premenné. Sú to premenné, ktoré môžeme používať v priebehu celého programu, či už ako argumenty funkcií alebo im budeme pripisovať návratovú hodnotu z funkcie. Výhodou týchto premenných je, že hodnota ktorá sa im priradí, sa zachováva až do momentu, kým nie je prepísaná, bez ohľadu na to, či program bežiaci v slučke skončil alebo nie<sup>2</sup>.

Typ a počet globálnych premenných závisí od programára, ale rád by som upozornil na to, že niektoré funkcie, ktoré sú zodpovedné napr. za časovanie alebo kalibráciu vyžadujú, aby pre ne bola vytvorená globálna premenná konkrétneho typu:

```
int Minimum;
int Maximum;
unsigned long Ts=688;
bool next=false; //zodpoveda za vzorkovanie
```

Kalibračná funkcia určuje hranice, v ktorých je magnet schopný lietať a dá sa povedať že ciachuje halovu sondu, aby sa ňou dala ovládať výška levitácie. Kedže tieto hranice sú pomerne často využívanou informáciou, ich hodnota si vyžaduje globálne premenné *Maximum* a *Minimum*. V inom prípade by ich nebolo možné využívať ako argumenty iných funkcií.

V prípade môjho programu som na vzorkovanie (určenie, ako často sa má program, ktorý načíta a spracuje výstupy a vytvorí vstupy do sústavy, zopakovať) použil funkcie vytvorené v rámci projektu AutomationShield. Aj tie vyžadujú globálne premenné, kedže sú využívané vrámcí celého programu ako argumenty.

Prvou takouto premennou je *Ts*, ktorá nesie hodnotu periódy vzorkovania. Ako som meraniami zistil, časť programu, ktorá beží v cykle, trvá 688  $\mu$ s a kedže na plynulosť levitácie okrem iného vplýva aj rýchlosť vzorkovania, nastavil som ju na čo najvyššiu, teda 688  $\mu$ s (pre prípad inej konštrukcie programu a jeho vzorkovania, kedy je využívaná aj sériová komunikácia, vidieť celý program v prílohe C).

Druhou premennou je boоловská premenná *next*. To, že je to boоловská premenná, znamená, že môže nadobúdať jedine hodnoty 1 (true) alebo 0 (false). Táto premenná je využitá v rámci funkcie definovanej na konci programu.

```
void stepEnable(){
    next=true;}
```

Po uplynutí *Ts=688  $\mu$ s* má za úlohu spustiť cyklickú časť programu odznovu. Táto funkcia sa nazýva interrupt a je špecifická tým, že akonáhle je nejakým dejom zavolaná, všetky funkcie sa zastavia, vykoná sa interrupt a až potom pokračuje zvyšok kódu.

Túto vlastnosť jej ale treba priradiť. To sa deje v časti *setup* pomocou funkcie, ktorej vstupným argumentom nie je premenná, ale celá funkcia:

```
Sampling.setInterruptCallback(stepEnable);
```

<sup>1</sup>Obsah tejto knižnice je tvorbou kolektívu pracujúceho na školskom projekte AutomationShield [3].

<sup>2</sup>Lokálne premenné, ktoré sa definujú vnútri programu sa s každým cyklom, v ktorom program prebehne, definujú nanovo a nie sú schopné si svoju hodnotu uchovávať.

Za deklaráciou premenných nasleduje časť programu, ktorá sa volá `void setup()`. Všetky funkcie a príkazy zadané v tejto časti sa zrealizujú iba raz, pri spustení celého programu.

V tejto časti okrem už spomenutej funkcie nájdeme príkazy, ktoré vykonajú inicializáciu, kalibráciu a nastavenie požadovanej výšky levitácie pre levitátor Magneto. Taktiež tu nájdeme funkcie, ktoré priradujú získané hranice z kalibrácie globálnym premenným, nastavenie konštánt PID regulátora a príkaz pre začatie sériovej komunikácie. Pomocou okna sériovej komunikácie si môžme vykresliť získané a zasielané údaje na Magneto. Táto komunikácia však veľmi spomaluje priebeh celého programu a tým pádom je levitácia nestabilnejšia. Z tohto dôvodu som sériovú komunikáciu síce inicializoval a napísal som aj kód na vykreslenie dát, ale sú zakomentované (v prípade potreby sa dajú odkomentovať a použiť).

Niekoľko krát som už spomenul "cyklická časť kódu". Je to časť programu, ktorá sa nachádza vo vnútri slučky `void loop()`. Slučkou sa myslí funkcia, ktorá je vykonávaná programom stále dookola, pokiaľ program beží. Na začiatku tejto funkcie sa nachádza podmienka, ktorá keď je splnená (premenná `next = 1` resp. `true`) prebehne funkcia `step()` a následne sa premennej `next` pridelí hodnota `0` resp. `false`.

Teraz trochu priblížim obsah funkcie `step()`. V prípade, že nepoužívame sériovú komunikácu, je naozaj krátka, obsahuje tri funkcie:

```
void step(){
    error = MagnetoShield.error();
    input=PIDAbs.compute(error,150,255,Minimum,Maximum);
    MagnetoShield.setVoltage(input);
}
```

O prvej a poslednej funkcií som už písal v Kap. 4.1.2. Sú zodpovedné za výpočet regulačnej odchýlky a za zaslatie získaného vstupu z PID regulátora do systému Magneto.

Stredná funkcia predstavuje číslicový regulátor PID. Je to kľúčová časť celého programu, zodpovedná za analýzu výstupov a vytváranie vstupov do sústavy. Označenie *PIDAbs* vyjadruje, že sa jedná o absolútny tvar PID algoritmu. Tento tvar počíta celkový vstup do sústavy  $u(kT)$ . Druhý spôsob, ako možno zapísať algoritmus PID je takzvaný prírastkový algoritmus PID, ktorý počíta len prírastok vstupu  $\Delta u(kT)$ .

Ide o funkciu s návratovou hodnotou typu float a piatimi vstupnými argumentmi. Prvý argument je regulačná odchýlka, ktorej vznik a význam som opísal v Kap. 5.1 (opisuje rozdiel požadovanej a momentálnej hodnoty regulovanej veličiny).

Ďalšie dva argumenty predstavujú minimálnu a maximálnu hodnotu vstupu. Ako som už spomenul, vstup do sústavy  $u(kT)$ , v prípade regulácie na shielde Magneto, predstavuje 8-bitové číslo zasielané DA prevodníku, na základe ktorého je generované napätie 0-5 V. Problém spočíva v tom, že PID regulátor nevie, aký najväčší (resp. najmenší) vstup do sústavy môže poslať. V prípade veľkej dlhotrvajúcej odchýlky  $e(kT)$  môže regulátor vyhodnotiť, že pošle do sústavy vstup o obrovskej hodnote, ktorú akčný člen nie je schopný vykonať. V prípade Magneta by to mohli byť vstupy v intervaloch  $(-\infty, 0) \cup (255, \infty)$ , čo by viedlo ku chaotickému správaniu sa prevodníka a aj celej sústavy. Preto je potrebné pre PID regulátor ohraničiť taký interval hodnôt, v akom je schopný akčný člen pracovať.

Ako možno vidieť, ohraničenie, ktoré som zadal PID regulátoru, nepredstavuje hodnoty 0 až 255, s ktorými je DA prevodník schopný pracovať, ale použil som ešte užší interval hodnôt. Je to z dôvodu, že DA prevodník ovplivňuje sústavu pomocou MOSFET-u, ktorého priepustnosť závisí od jeho parametrov. Meraniami som dprišiel k záveru, že až

napäťie približne 2,8 V otvára MOSFET IRF520 a tomuto napätiu zodpovedá 8-bitová hodnota 155. Vďaka tomuto poznatku som mohol zmenšiť interval vstupov na  $\langle 155, 255 \rangle$  a tým zlepšiť plynulosť riadenia. Hodnoty, ktoré som použil boli získané čisto experimentálne a pre dve rôzne sústavy sa môžu trochu lísiť.

Posledná dvojica vstupov sú parametre proti vynášaniu hodnôt, takzvanému “wind-up” efektu. Ide o jav spojený s integračnou časťou PID regulátora. Ako som uviedol v Kap. 5.1, integračná časť PID akumuluje odchýlku počas celej doby regulácie. V prípade prílišného nárastu tejto zložky, akonáhle sa dosiahne požadovaná hodnota, regulovaná veličina bude ďalej narastať a až po určitom čase nad požadovanou hodnotou, začne klesať. To že meraná veličina musí vystúpať nad požadovanú hodnotu je dané tým, že odchýlka s kladným znamienkom nahromadená počas doby dosahovania požadovanej hodnoty v integrály (alebo v sume v diskrétnej forme regulátora), sa musí vyrovnať odchýlkou s opačným znamienkom (vypíva z Rov. (5.4)). Tento efekt je samozrejme nechcený, lebo viedie k rozkmitávaniu sústavy a predĺžuje dobu ustálenia sa na požadovanej hodnote.

Existuje viacero spôsobov ako tomuto efektu zabrániť, alebo ho v čo najväčšej miere obmedziť. Jeden zo spôsobov, ktorý je použitý aj v tejto funkcií PID regulátora, je obmedzenie hodnôt, ktoré môže integračná zložka nadobudnúť zadaním hodnôt *Minimum* a *Maximum*. Vďaka tomu nemôže jej veľkosť narastať do nekonečna a aj prekmitnutie nad požadovanú hodnotu bude menšie a v prípade správne vyladeného PID by nemalo nastať vôbec.

Pre výpočet “anti-wind-up” hodnôt, ktoré ohraničujú činnosť integračnej zložky je požitá nasledovná nerovnosť:

$$\text{Minimum} < (K_p * T_s / T_i) * eSum < \text{Maximum} \quad (5.8)$$

kde  $eSum$  predstavuje súčet všetkých nameraných regulačných odchýľok  $e(kT)$ .

Súčasťou ukážkového príkladu sú ďalšie dva varianty funkcie `step()`. Principiálne sa jedná o tú istú funkciu, len v týchto variantoch je do programu začlenená aj sériová komunikácia, vďaka ktorej je používateľ schopný zapísat a vykresliť hodnoty vstupov a výstupov zo sústavy v reálnom čase.

V oboch prípadoch som nepoužil fukciu `MagnetoShield.error()`, ale zosnímanie pozície magnetu, ako aj odčítanie od požadovanej hodnoty levitácie som vykonal priamo v kóde (viď C). Urobil som tak preto, lebo pri vykreslení vstupno-výstupnej charakteristiky je potrebné mať samostatne uloženú premennú okamžitej pozície a okamžitého vstupu. Tieto premenné sa potom vykreslujú pomocou príkazov:

```
Serial.print(output); //vypise text
Serial.print(" "); //urobi medzeru
Serial.println(input); //vypise text a zalomi riadok
```

Druhá zakomentovaná varianta vykresluje vstupno-výstupnú charakteristiku v percentách. Preto sú premenné *output* a *input* po nadobudnutí hodnoty ešte prekonvertované na percentá pomocou funkcií:

```
float outputPer = AutomationShield.mapFloat(output, Minimum, Maximum, 0.00, 100.00);
float inputPer = AutomationShield.mapFloat(input, 150.00, 255.00, 0.00, 100.00);
```

Vykreslovanie vstupov a výstupov v percentoch je praktickejšie ako zobrazovanie ich reálnych hodnôt, pretože používateľovi priamo ukazujú, aký je všťah medzi výkonom akčného člena a zmenou sústavy, bez toho aby musel brať ohľad na fyzikálny rozmer premenných.

V prípade potreby možno tieto formy funkcie `step()` odkomentovať a použiť, no treba počítať s tým, že využívanie sériovej komunikácie výrazne predĺži čas programu a na základe toho treba upraviť aj vzorkovanie<sup>3</sup>. To vedie k výrazne nestabilnejšej regulácií levitácie, čo sa odrazí aj na kvalite získaných údajov. Z tohto dôvodu odporúčam využívať iné metódy vykreslovania vstupno-výstupnej charakteristiky namiesto sériovej komunikácie.

### 5.3 Určovanie konštánt a práca s levitátorom Magneto

Získanie grafickej informácie o tom, ako vstupná veličina ovplyvňuje výstupnú a akým spôsobom systém dosahuje požadovanú hodnotu, je nutnou podmienkou efektívnej práce so systémom a jeho správneho nalaďenia.

Jeden zo spôsobov, ako možno vykresliť vstupno-výstupnú charakteristiku bez použitia sériovej komunikácie je pomocou osciloskopu. Týmto spôsobom nedostaneme vzájomný pomer vstupov a výstupov v percentách, ale v podobe meniaceho sa napätia. Kedže požadovaná hodnota je zadávaná v percentách a mikropočítač spracováva údaje o napätií na hallovom senzore vo forme 10-bitového čísla, je potrebné vykonať prepočet na zistenie požadovanej hodnoty levitácie vo voltoch V.

V prípade názorného príkladu som pomocou funkcie `MagnetoShield.setHeight(50.00)` zvolil za požadovanú hodnotu levitácie  $Per = 50\%$  z celkovej výšky v ktorej sa môže magnet pohybovať. Aby som vedel systém vyladiť, musím zistiť, akému výstupnému napätiu sa to rovná. To môžem zistiť tak, že pri kalibrácii Magneta budem sledovať na osciloskope výstupné napätie zo systému a vezmem jeho najvyššiu hodnotu  $MaxV$  a najnižšiu hodnotu  $MinV$ . Potom hľadaná hodnota *Hodnota* bude:

$$Hodnota = \frac{MaxV - MinV}{100} * Per + MinV \quad (5.9)$$

Druhý spôsob ako je možné zistiť hľadanú hodnotu je pomocou funkcií

```
int MagnetoShield.getMin()
int MagnetoShield.getMax()
```

ktoré vracajú 10-bitovú hodnotu maxima *Max* a minima *Min* zistených pri kalibrácii. Následne možno vypočítať napätie zodpovedajúce požadovanej polohe magnetu pri levitácií ako:

$$Hodnota = \frac{4.7 - 0.2}{1023} * \frac{Max - Min}{100} * Per \quad (5.10)$$

Rozdiel  $4.7 - 0.2$  udáva, aké výstupné napätie vo voltoch z hallovho senzora zodpovedá 0 a aké zodpovedá hodnote 1023 (interval v akom mikrokontroler spracúva namerané napätie). Tieto hodnoty sú získané z datasheetu [1].

Pri ladení sústavy som používal USB osciloskop Analog Discovery 2 (Obr. 5.2), ktorý má okrem iného aj možnosť merať napätie so vzorkovaním s frekvenciou 8 kHz. Zariadenie je pripojiteľné k počítaču, vďaka čomu je možné namerané hodnoty zobraziť a ďalej spracovať.

Pomocou tohto zariadenia som meral vplyv jednotlivých konštánt na sústavu (ako sa mení výstupné napätie) a snažil som sa dosiahnuť takú kombináciu, aby levitujúci magnet

---

<sup>3</sup>Odporučané časi vzorkovania pre jednotlivé varianty funkcie `step()` uvádzam na začiatku programu.



Obr. 5.2: USB osciloskop Analog Discovery 2.

dosiahol požadovanú výšku levitácie, vystúpal do nej v čo najkratšom čase a po dosiahnutí požadovaného stavu pokračoval v levitácií pokiaľ možno čo najrovnomernejšie.

Pri určovaní konštánt PID regulátora som sa vzhľadom na komplexnosť systému snažil postupovať metódou pokusov a opráv. Na Obr. 5.3 je požadovaná hodnota lietania znázornená zelenou farbou, napätie odčítané z hallovho senzora (výstup) červenou a napätie generované DA prevodníkom modrou (vstup).

Jednou z vecí, ktorú si mňo všimnút<sup>4</sup>, je to, že napätie vytvárané DA prevodníkom sa pohybuje približne v intervale 2.8 až 4.2 V. Spodná hranica je daná ohraničením, ktoré sme vytvorili vo funkcii `PIDAbs.compute(error, 155, 210, Minimum, Maximum)`. Vrchná hranica je ovplyvnená LED diódou, ktorá je napájaná prevodníkom. Ako som spomíнал, DA prevodník by mal byť schopný generovať napätie 0 až 5 V, avšak z dôvodu pripojenej LED v obvode, ako aj prípadne iných napäťových strát je táto hodnota o niečo nižšia (napájanie DA prevodníka nie je rovných 5 V - aj to vplýva na generované napätie). Experimentálne som zistil, že v prípade náhrady rezistora pred LED diódou za rezistor s odporom  $1.2\text{ k}\Omega$ , generované napätie stúpnena približne 4.2 V, čo je postačujúce pre otvorenie MOSFETU na prepúšťanie 12 V pre napájanie elektromagnetu (v prípade potreby čo najviac voltov z DA prevodníka, je najlepšie LED do obvodu vôbec nezapájať).

Ako prvé som určil konštantu  $K_p$  pri vyradenej integračnej zložke ( $T_d \rightarrow \infty$ ) a derivačnej zložke ( $T_i \rightarrow 0$ )<sup>4</sup>. Akonáhle som dosiahol takú hodnotu  $K_p$ , pri ktorej sa magnet odlebil od zeme - v pravidelných intervaloch vylietal do maximálnej polohy a padal na späť, zaraďil som derivačnú zložku  $T_d$ , ktorú som postupne zvyšoval. Ako možno vidieť na

---

<sup>4</sup>Na základe Rov. (5.5) pre vyradenie integračnej zložky musí byť konštanta  $T_i$  čo najväčšia a pre vyradenie derivačnej zložky musí byť konštanta  $T_d$  naopak čo najmenšia.

Obr. 5.3(a), síce sa magnet dostane do blízkosti požadovanej hodnoty, ale z dôvodu príliš malej derivačnej zložky dochádza k tlmenému kmitaniu a z dôvodu vyradenej integračnej zložky nastáva trvalá regulačná odchýlka.

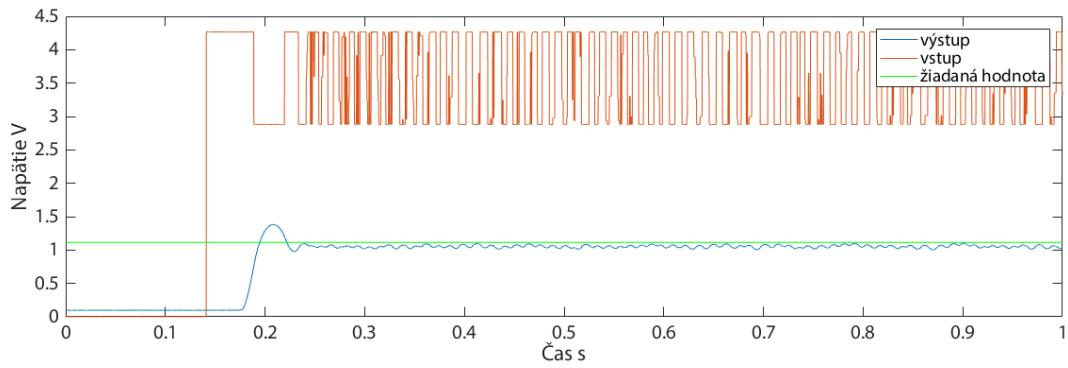
Pokračoval som vo zvyšovaní časovej derivačnej  $T_d$  konštanty. Tým som dosiahol, že sústava neprekmitala vôbec, ale trvalá regulačná odchýlka bola stále výrazná (Obr. 5.3(b)).

Nakoniec som znížil koeficient  $T_i$  a zaraďil som do regulácie aj integračnú zložku, ktorá je zodpovedná za odstránenie trvalej regulačnej odchýlky (Obr. 5.3(c)). Výsledkom je pomerne rýchle nadobudnotie požadovanej hodnoty s malou odchýlkou pri relatívnej stabilité levitácie.

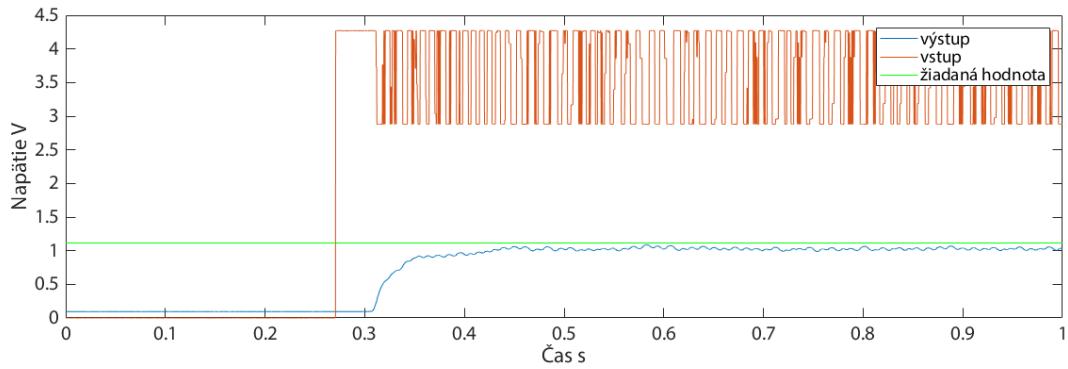
Akým spôsobom sa prejavuje nízka hodnota  $K_p$  v systéme predstavuje Obr. 5.3(d).

Ešte raz by som rád zdôraznil, že každý levitátor Magneto predstavuje osobitý systém. Vďaka dynamickému deju, ktorý je ním realizovaný, sa nemusia na všetkých levitátoroch dosiahnuť rovnaké výsledky a vyššie uvedený príklad by sa mal chápať skôr ako ukážka, akým spôsobom postupovať pri práci s Magnetom, než ako všeobecne aplikovatelný program.

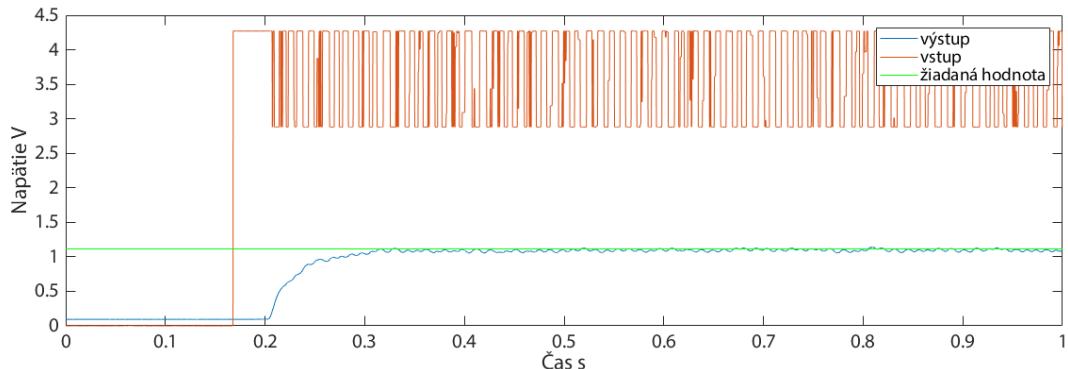
Zároveň na Obr. 5.3 a 5.3(d), vstup do sústavy znázorňený červenou farbou, môže pôsobiť ako vstup generovaný On-OFF riadením. Tento charakter zmeny vstupného napäťia je daný veľkým počtom výpočtov a na základe nich vytvorených vstupov behom jednej sekundy (vzorkovanie má pri perióde  $688 \mu\text{s}$  frekvenciu viac ako  $1.4 \text{ kHz}$ ). Na Obr. 5.4 možno vidieť, že sa o ON-OFF riadenie nejedná a vstupné napätie nadobúda aj iné hodnoty, ako sú minimum a maximum. Zároveň je znázornené, ako ovplyvňuje výstupné napätie charakterizujúce polohu magnetu.



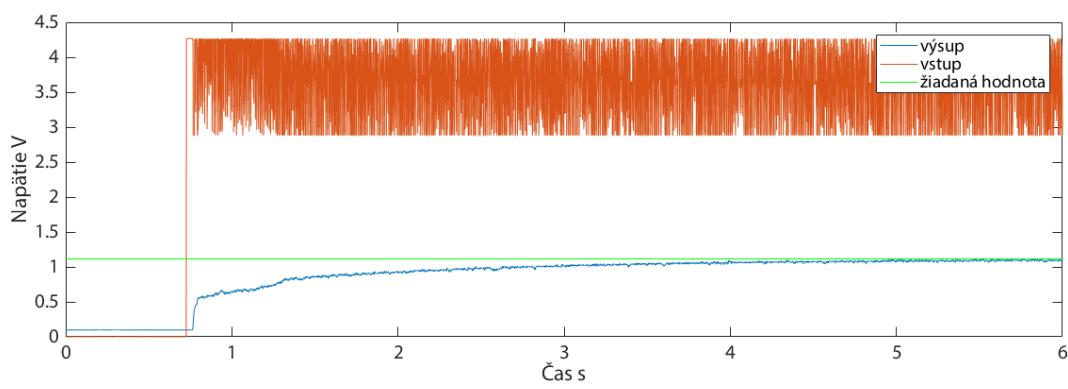
(a)  $K_p = 16.6, T_i = 30, T_d = 0.005$



(b)  $K_p = 16.6, T_i = 30, T_d = 0.02$

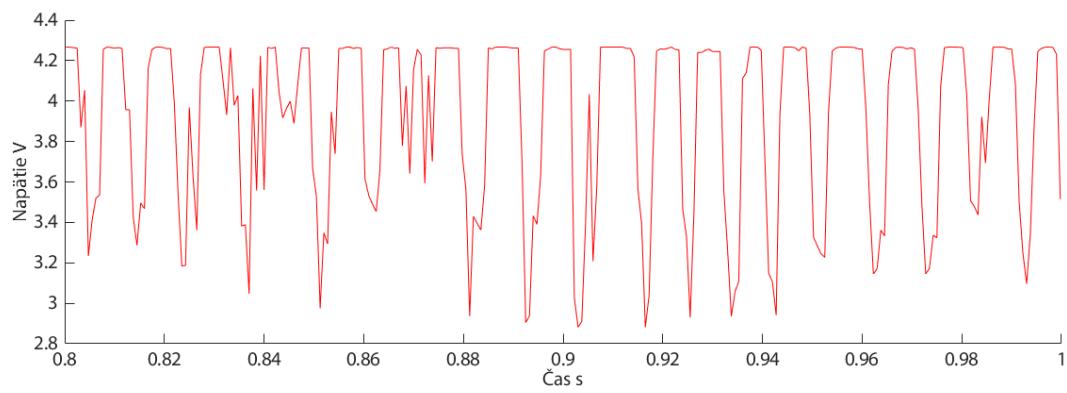


(c)  $K_p = 16.6, T_i = 1.5, T_d = 0.02$

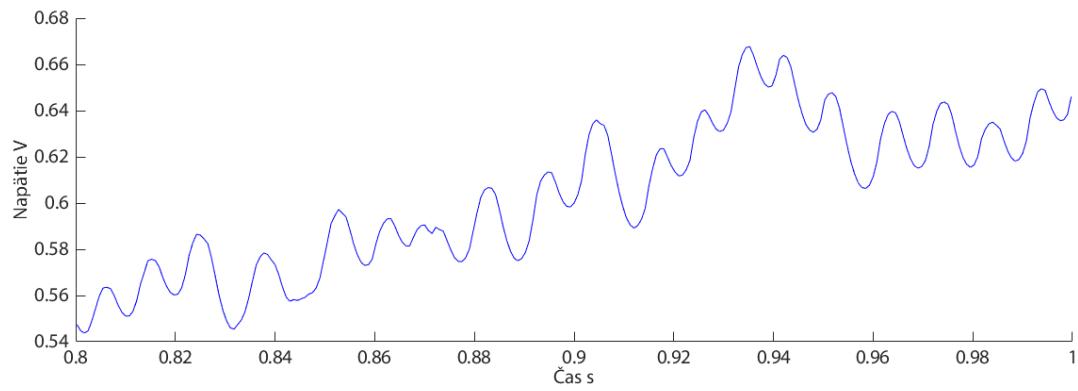


(d)  $K_p = 16.6, T_i = 1.5, T_d = 0.02$

Obr. 5.3: Postup určovania konštant PID regulátora



(a) Vstupné napätie.



(b) Výstupné napätie.

Obr. 5.4: Detail vstupného a výstupného napäťia z Obr. 5.3(d).

# Záver

Príkladom v poslednej kapitole som preukázal, že miniaturizácia elektromagnetického levitátora do podoby shieldu bola úspešná, nakoľko magnet v levitátore je schopný lietať. Na Obr. 5.3 je vidieť, že sústava reaguje na zmenu nastavení regulačného algoritmu, čo demonštruje možnosť využitia tohto zariadenia ako didaktickej pomôcky.

Napriek skonštatovaniu úspešnej realizácie, levitátor má určité obmedzenia, ktoré pri práci s ním treba vziať do úvahy.

Prvým obmedzením je už spomenutá prieplustnosť MOSFET-u. DA prevodník otvára MOSFET až pri napätí približne 2.8 V. Zmenšenie intervalu vstupných hodnôt sice priebeh regulácie systému nijako výrazne neovplyvní, ale je potrebné s ním rátat' pri práci, návrhoch a prípadne ďalšom vývoji levitátora Magneto.

Ako je zrejmé z Obr. 5.3 a 5.3(d), pri levitácií dochádza k miniaturnému kmitaniu magnetu, ktoré nie je dôsledkom vyladenia sústavy. Toto kmitanie môže byť spôsobené viacerými faktormi.

Magnetické poľe, ktoré elektromagnet vytvára, sa môže meniť, nie len na základe zvislej vzdialenosťi od povrchu elektromagnetu. Ak by sme merali intenzitu poľa v rámci roviny kolmej na os elektromagnetu, s tým že merania by sme robili zakaždým na inom mieste, ale v rovnakej vzdialenosťi od osi elektromagnetu (merané body by vykreslili kružnicu), pravdepodobne by sme namerali rôzne hodnoty. Tento predpoklad vyplýva z pozorovaní, keď elektromagnet príťahoval magnet, trajektória zakaždým viedla mimo osi elektromagnetu (magnet nevyletel rovno hore, ale najprv sa trochu posunul do strany mimo os elektromagnetu a až potom sa prisal). Tento jav môže byť spôsobený výrobnou nedokonalosťou elektromagnetu.

Ďalšou príčinou kmitania sústavy môže byť meranie polohy hallovým senzorom. Ako vidno z predchádzajúcich kapitol, zisťovanie polohy týmto spôsobom je účinné a pre riadenie levitácie dostačujúce. Hallov senzor však nemeria polohu priamo a prípadné vychýlenia magnetu, ktorého magnetické pole meria, môžu vytvárať nesprávne informácie o polohe a to spôsobuje kmitanie. Prípadne aj magnetické pole elektromagnetu môže byť zaznamenané hallovým senzorom a to sa prejavuje na grafoch vykreslením kmitavého pohybu

Najpravdepodobnejším dôvodom týchto kmitov je nemožnosť dosiahnuť dokonalé podmienky levitácie PID regulátorom. Na to, aby teleso levitovalo v pravom slova zmysle musí byť splnené  $\vec{F}_m = -\vec{F}_G$  a  $\vec{v} = 0$  (vid' Kap. 1.1). To však PID regulátor pre komplikovanosť tohto dejia celkom nedokáže. Z tohto dôvodu by som odporučil na danú sústavu aplikovať iné komplexnejšie spôsoby riadenia, avšak v rámci možností riadiaceho mikročipu ATmega328P.

Po skonštruovaní shieldu a pri overovaní jeho funkčnosti som dospel k niekoľkým

dizajnovým zmenám, ktoré by v prípade opäťovného návrhu takéhoto levitátora mali byť zvážené.

Na I<sup>2</sup>C komunikáciu Magneto využíva piny A4, A5. Nevýhodou tohto riešenia je, že shield Magneto je kompatibilný len s prototypizačnou doskou Arduino UNO. V prípade opäťovného návrhu odporúčam použiť piny, ktoré sa nachádzajú na shielde vedľa pinu AREF (vid Obr. 3.1). Tým pádom by shield bol kompatibilný aj s doskou Arduino MEGA.

Konštrukciu pre uchytenie elektromagnetu tvoria 4 dištančné stĺpiky, aby nedošlo k rozkmitávaniu hliníkovej pásoviny, ku ktorej je magnet prichytený. Po skonštruovaní levitátora sa ukázalo, že aj 2 stĺpiky len na jednej strane predstavujú dostatočnú oporu a pozorovanie levitácie je navyše prehľadnejšie.

Vyššie uvedené nedokonalosti dizajnu, ako aj neschopnosť dokonale ustálenej levitácie, nie sú veci znemožňujúce primárny cieľ tohto zariadenia. Skôr by sa mali chápať ako nárazy na vylepšenie a zdokonalenie už fungujúceho produktu.

# Literatúra

- [1] Allegro MicroSystems. Continuous-time ratiometric linear hall effect sensors. Online datasheet, Allegro MicroSystems, LLC, 2013. [cit. 2018-04-24], <http://robotics.itee.uq.edu.au/~elec3004/laboratories/LeviLab/Hall%20Effect%20Sensor%20A1301-2-Datasheet.pdf>.
- [2] ARDUINO. Wire library. Online., 2018. [cit. 2018-05-16], <http://www.mathworks.com/help/ident/ug/algorithms-for-recursive-estimation.html>.
- [3] AutomationShield. Automationshield repository. online GitHub, 2018. [cit. 2018-05-20], <https://github.com/gergelytakacs/AutomationShield>.
- [4] C. Belavý. *Základy automatizácie a merania*. Slovenská technická univerzita v Bratislave, Bratislava, Vazovova 5, 2012.
- [5] C.R.Nave. Resistance: Temperature coefficient, low temperature resistivity, resistor temperature dependence. Georgia State University website HyperPhysics. Online., 2000. [cit. 2018-04-22], <http://hyperphysics.phy-astr.gsu.edu/hbase/electric/restmp.html>.
- [6] P. H. "Hawelson". Levitátor. Online návod., 2006. [cit. 2018-04-21], <http://hawelson.blog.cz/0609/levitator>.
- [7] J. Liberty and B. L. Jones. *Sams Teach Yourself C++ in 21 Days*. Computer Press, Sams Indianapolis, IN, USA, 2007.
- [8] K. Magnetics. Electromagnetic levitation. Online. [cit. 2018-04-21], <https://www.kjmagnetics.com/blog.asp?p=electromagnetic-levitation>.
- [9] R. Marston. Fet principles and circuits — part 1. NUTS & VOLTS MAGAZINE. Online., Máj 2000. [cit. 2018-04-23], [http://www.nutsvolts.com/magazine/article/fet\\_principles\\_and\\_circuits\\_part\\_1](http://www.nutsvolts.com/magazine/article/fet_principles_and_circuits_part_1).
- [10] MATLAB. Introduction: Pid controller design. Online. [cit. 2018-05-16], <http://ctms.engin.umich.edu/CTMS/index.php?example=Introduction&section=ControlPID>.
- [11] NXP Semiconductor. Pcf8591. Online datasheet, NXP Semiconductors, 2013. [cit. 2018-04-24], <https://www.mouser.sk/datasheet/2/302/PCF8591-1127459.pdf>.
- [12] Quanser. Magnetic levitation. Online datasheet. [cit. 2018-04-21], <https://www.quanser.com/wp-content/uploads/2017/03/MAGLEV-Datasheet.pdf>.

- [13] J. Reichl and M. Všetička. Princip činnosti tranzistoru s indukovaným kanálem. Encyklopédie fyziky. Online. [cit. 2018-04-23], <http://fyzika.jreichl.com/main/article/view/1669-princip-cinnosti-tranzistoru-s-indukovanym-kanalem>.
- [14] A. Tirpák. *Elektromagnetismus*. IRIS, Bratislava, Slovakia, 2014.
- [15] Tutorials Point. C - header files. Online programovací tutoriál. [cit. 2018-05-15], [https://www.tutorialspoint.com/cprogramming/c\\_header\\_files.htm](https://www.tutorialspoint.com/cprogramming/c_header_files.htm).
- [16] Vishay Siliconix. Irf520, sihf520. Online datasheet 91000, Vishay Intertechnology, Inc., 2017. [cit. 2018-04-23], <https://www.vishay.com/docs/91017/91017.pdf>.
- [17] A. Yadav. Effect of temperature on electric current, magnets and electromagnet. International Journal of Advancements in Technology. Online., 14.7.2016. [cit. 2018-04-22], <https://www.omicsonline.org/open-access/effect-of-temperature-on-electric-current-magnets-and-electromagnet-0976-4860-1000167.php?aid=75409>.

## A Zdrojový kód *header* súboru

```
#ifndef MagnetoShield_h           //ochrana pred viacnasobnym pridanim
#define MagnetoShield_h

#include "AutomationShield.h" //zahrnie kniznicu utomationShield
#include "Wire.h"             //potrebne pre DA prevodnik

#define PCF8591 (0x90 >> 1)
#define HallSenzor A3

class MagnetoShieldClass        //deklaracia triedy
{
public:

    void begin();                //inicIALIZacia pinov + zahrnie
                                  ////kniznicu "Wire"

    uint8_t setVoltageV(float voltagePer); //napatie 0-5 V na DA prevodniku
                                         ////vracia 8-bitove cislo

    float readHeight();          //okamzita hodno hallovho senzoru

    float setHeight(float set);   //nastavi poziciu lietnia v %
                                  ////a vrati 10-bitove cislo

    void calibration();          //nakalibruje levitator Magneto

    int getMin();                 //vracia 10-bitovu hodnotu
                                  //najnizsiej pozicie magnetu

    int getMax();                 //vracia 10-bitovu hodnotu
                                  //najvyssej pozicie magnetu

    float error();                //rozdiel pozadovanej a chcenej
                                  ////polohy lietania -> pre PID

    void setVoltage(float u);     //posle napatie na DA prevodnik
                                  ////8-bitove cislo 0=>0V; 255=>5V

private:

    float Min;                  //10-bitova hodnota minimalnej polohy lietania
    float Max;                  //10-bitova hodnota maximalnej polohy lietania
    float setpoint; //pozadovana hodnota lietania
};

extern MagnetoShieldClass MagnetoShield; //ako budu pirkazi volane externe
#endif
```

## B Zdrojový kód *source* súboru

```
#include "MagnetoShield.h" //prepojenie s header suborom

void MagnetoShieldClass::begin()
{
    pinMode(HallSenzor, INPUT); //nastavi HallSenzor (pin) ako
                                //INPUT (vstup)
    Wire.begin();             //inicjalizuje "Wire" kniznicu
}

uint8_t MagnetoShieldClass::setVoltageV(float voltage)
{
    if(voltage > 5.00){                  //ohranicenie intervalu 0.00-5.00
        voltage = 5.00;}
    else if (voltage < 0.00){
        voltage = 0.00;}
    float voltageF = voltage*255.00/5.00; //konverzia intervalu
                                            //na 0.00-255.00
    uint8_t voltageB = (uint8_t) voltageF; //konverzia float na 8-bit
    Wire.beginTransmission(PCF8591);       //oslovenie PCF cez adresu
    Wire.write(0x40);                     //zvolenie funkcie DA prevodniku
    Wire.write(voltageB);                //hodnota generovaneho napatia
    Wire.endTransmission();              //ukoncenie I2C komunikacie
    return voltageB;                   //navratova hodnota funkcie
}

void MagnetoShieldClass::calibration()
{
    Wire.beginTransmission(PCF8591); //I2C komunikacia
    Wire.write(0x40);
    Wire.write(0);
    Wire.endTransmission();
    delay(1000);                  //caka na vytvorenie podmienok
    Min = analogRead(HallSenzor); //najnizsia poloha magnetu

    Wire.beginTransmission(PCF8591); //I2C komunikacia
    Wire.write(0x40);
    Wire.write(255);
    Wire.endTransmission();
    delay(1000);                  //caka na vytvorenie podmienok
    Max = analogRead(HallSenzor); //najvyssia poloha magnetu

    Wire.beginTransmission(PCF8591); //nastavenie vzletu na zem
    Wire.write(0x40);
    Wire.write(0);
    Wire.endTransmission();
    delay(1000);
}
```

```

int MagnetoShieldClass::getMin()
{
    int variable = (int) Min; //konverzia float->int
    return variable;
}

int MagnetoShieldClass::getMax()
{
    int variable = (int) Max; //konverzia float->int
    return variable;
}

float MagnetoShieldClass::readHeight()
{
    float Height = analogRead(HallSenzor); //momentalna poloha lietania
    return Height;
}

float MagnetoShieldClass::setHeight(float set)
{
    //prepočet polohy lietania z %->float v intervale <Min,Max>
    setpoint = AutomationShield.mapFloat(set,0.00,100.00,Min,Max);
    return setpoint;
}

void MagnetoShieldClass::setVoltage(float u)
{
    uint8_t u8B = (uint8_t) u;           //konverzia float->uint8_u
    Wire.beginTransmission(PCF8591); //I2C komunikacia
    Wire.write(0x40);
    Wire.write(u8B);
    Wire.endTransmission();
}

float MagnetoShieldClass::error()
{
    float position = analogRead(HallSenzor); //momentalna pozicia magnetu
    float err = setpoint-position;           //setpoint funkcie setHeight()
    return err;
}

MagnetoShieldClass MagnetoShield;           //meni nazov MagnetoShieldClass
                                         //na MagnetoShield

```

## C PID regulácia Magneta

```
#include <AutomationShield.h>

float input, error, Setpoint, output; //deklaracia globalnych premennych
                                         //pre .ino program
int Minimum;
int Maximum;

unsigned long Ts=688;                  //preioda vzorkovania v mikrosekundach
/*
 * Casy vzorkovania pre rozne typy programu:
 * 1) pre levitaciu bez seriovej komunikacie = 688 microsekund
 * 2) lietanie s komunikaciou = 1536 microsekund
 * 3) lietanie s komunikaciou a pemennou vstupov a vystupov na % = 1476
   microsekund
 */

bool next=false;                      //premenna spustajuca step() funkciu

void setup() {
    Serial.begin(230400);              //rychlosť seriovej kom. v baudoch [bity/s]

    MagnetoShield.begin();
    MagnetoShield.calibration();
    Setpoint = MagnetoShield.setHeight(50.00); //vracia privatnu premennu setpoint
                                                //bez seriovej komunikacie
                                                //Setpoint=... nepotrebné

    Minimum=MagnetoShield.getMin();      //zistenie hranic lietania
    Maximum=MagnetoShield.getMax();

    Sampling.initialize(Ts);            //nastavenie vzorkovania
    Sampling.setInterruptCallback(stepEnable); //co sa ma stat
                                                //ked sa zavola interrupt

    PIDAbs.setKp(12);                 //nastavenie PID konstant
    PIDAbs.setTi(1.5);
    PIDAbs.setTd(0.02);

}

void loop() {                         //vykona funkciu step() ak next=true
    if(next){
        step();
        next=false;
    }
}
```

```

void stepEnable(){ //obsah interruptu -> povoli step()
    next=true;
}

void step(){
// 1) Lietanie bezseriovej komunikacie - najplinulejsia regulacia:
    error = MagnetoShield.error();           //vypocet regulacnej odchylky e(t)
    input=PIDAbs.compute(error,155,255,-65000,65000); //PID regulacia - hodnoty
                                                    //zavisia od sustavy
    MagnetoShield.setVoltage(input);          //vnasa vstupy do systemu

// 2) Seriova komunikacia vykresluje vstupy a vystupy v hodnotach spracovavanych
    mikrocipom:
/*     output = MagnetoShield.readHeight(); //vystup zo systemu - pozicia magnetu
    error = Setpoint-output;                //rozdiel pozadovanej a realnej
                                            //pozicie lietania
    input=PIDAbs.compute(error,155,255,-65000,65000); //vypocet vstupov
    MagnetoShield.setVoltage(input);          //vnasa vstupy do systemu
    Serial.print(output);                  //vykresluje vstupy a vystupy v
                                            //seriovom okne
    Serial.print(" ");
    Serial.println(input);
*/
// 3) Seriova komunikacia vykresluje vstupy a vystupy v %:
/*     output = MagnetoShield.readHeight(); //vystup zo systemu - pozicia magnetu
    error = Setpoint-output;                //rozdiel pozadovanej a realnej
                                            //pozicie lietania
    input=PIDAbs.compute(error,155,255,-65000,65000); //vypocet vstupov
    MagnetoShield.setVoltage(input);          //vnasa vstupy do systemu
    //konverzia vystupov a vstupov na %
    float outputPer = AutomationShield.mapFloat(output,Minimum,Maximum
        ,0.00,100.00);
    float inputPer = AutomationShield.mapFloat(input,155.00,255.00,0.00,100.00);
    Serial.print(outputPer);                //zobrazi vstupy a vystupy v %
    Serial.print(" ");
    Serial.println(inputPer);
*/
}

```